

edv aspekte

2|89

Herausgegeben
von der Redaktion
rechentechnik
datenverarbeitung
DDR 5.00 M

**Verteilte
Verarbeitung**



Inhalt

<i>Prof. Dr. Gerd Rossa:</i> Rechnergestützte Betriebswirtschaft und verteilte Verarbeitung	2
<i>Prof. Dr. Claus Sattler:</i> Gestaltung netzfähiger Software für Arbeitsprozesse im Büro	5
<i>Prof. Dr. Gerd Rossa:</i> Datenbasis-Entwurfs-System DES	10
<i>Uwe Schulze:</i> Dienstorganisation in lokalen Netzen mittels Fernaufruf	14
<i>Prof. Dr. Gerd Rossa:</i> Kopplung von DBS und XPS Semantische Datenbeschreibung	17
<i>Axel Wüstemann:</i> Netzbetriebssysteme für den Verbund von Arbeitsplatzrechnern	18
<i>Uwe Schulze:</i> NETBIOS – standardisierter Netzzugang für PC-Technik	22
<i>Prof. Dr. Gerd Rossa:</i> Modell eines verteilten Datenbank- Betriebs-Systems (VDB)	25
<i>Uwe Schulze, Axel Wüstemann, Mat- thias Ohlerich:</i> Bürokommunikation mit Micro-NET-80	30
<i>Heidrun Ortleb:</i> Verteilte Verarbeitung – Datenkon- vertierung bei relationalen Datenban- ken	33
<i>Frank Försterling, Ingo Kerlikowski, Prof. Dr. Claus Sattler, Dr. Christine Tessmer:</i> Rechnergestützte Arbeit mit Mittei- lungen im Büro	35
<i>Dr. Gerald Hartung:</i> Auftragstransfer – Ein Anwendungs- dienst im ESER-Rechnerverbund	39
<i>Dr. Peter Erward:</i> DAFEMA-PC: Kopplungssoftware für die verteilte Verarbeitung	42
<i>Rolf Stickler, Axel Wüstemann:</i> NetWare Hochtechnologie bei PC- Netzen	46
<i>Dieter Lenz, Volker Nawrotzki, Martin Holzhauser</i> Zimmerreservierungssystem HODIS-2 unter Nutzung von SCOMLAN	50
<i>Axel Wüstemann:</i> Der Einsatz von lokalen Netzen in Betrieben	56
<i>Prof. Dr. Gerd Rossa, Prof. Dr. Martin Graef, Frank Bolz:</i> Drei Beiträge aus LO+ EDV '88	61

Содержание

<i>Проф. д-р Герд Росса:</i> Управление экономикой предпри- ятия с помощью ЭВМ и разделенная обработка	2
<i>Проф. д-р Клаус Заттлер:</i> Формирование приспособленного для сетевой работы матобеспечения процессов работы в бюро	5
<i>Проф. д-р Герд Росса:</i> Система проектирования базы дан- ных ДЕС	10
<i>Уве Шульце:</i> Организация службы локальной сети посредством телевызова	14
<i>Проф. д-р Герд Росса:</i> Стыковка ДБС и ХПС Семантическое описание данных	17
<i>Аксель Вюстеман:</i> Сетевые операционные системы для сопряжения автоматизированных ра- бочих мест	18
<i>Уве Шульце:</i> NETBIOS – стандартный доступ ПЭВМ к сети	22
<i>Проф. д-р Герд Росса:</i> Модель распределенной системы банка данных (ЖДБ)	25
<i>Уве Шульце, Аксель Вюстман, Матиас Олерих:</i> Учрежденческая коммуникация с помощью микроNET-80	30
<i>Хайдрун Ортелеб:</i> Распределенная обработка – прео- бразование данных в реляционных базах данных	33
<i>Франк Фёрстерлинг инго керликовски, проф. д-р Клаус Заттлер,</i> <i>д-р Кристине Тесмер:</i> Автоматизированная работа с сооб- щениями в учреждении	35
<i>Д-р Геральд Хартунг:</i> Преобразование заголовков – пользова- тельская служба в многомашинном комплексе ЕС ЭВМ	39
<i>Д-р Петер Эрвард:</i> ДАФЕМА-ПС: стыковочное матобес- печение для распределенной обра- ботки	42
<i>Рольф Штиклер, Аксель Вюстеман:</i> Нетваре – передовая технология у сетей ПЭВМ	46
<i>Дитер Ленц, Фолькер Навроцки, Мар- тин Хольцхауэр:</i> Система резервирования номеров ХОДИС-2 с использованием ССОМ- ЛАН	50

In this Issue

<i>Prof. Dr. Gerd Rossa:</i> Computerised enterprise management and distributed processing	2
<i>Prof. Dr. Claus Sattler:</i> Design of network-capable software for work processes in the office	5
<i>Prof. Dr. Gerd Rossa:</i> DES data base design system	10
<i>Uwe Schulze:</i> Service organisation in local networks by means fo remote call	14
<i>Prof. Dr. Gerd Rossa:</i> Coupling of DBS and XPS Semantic data description	17
<i>Axel Wüstemann:</i> Network operating systems for link- ing workstation computers	18
<i>Uwe Schulze:</i> NETBIOS–Standardised network access for PC hardware	22
<i>Prof. Dr. Gerd Rossa:</i> Model of a distributed data bank operating system (VDB)	25
<i>Uwe Schulze, Axel Wüstemann, Mat- thias Ohlerich:</i> Office communication based on MicroNET-80	30
<i>Heidrun Ortleb:</i> Distributed processing–Data conver- sion with relational data banks	33
<i>Frank Försterling, Ingo Kerlikowski, Prof. Dr. Claus Sattler, Dr. Christine Tessmer:</i> Computer-assisted work in the office	35
<i>Dr. Gerald Hartung:</i> Job transfer–Application service in ESER (USC)-based multiprocessor operation	39
<i>Dr. Peter Erward:</i> DAFEMA-PC: Coupling software for distributed processing	42
<i>Rolf Stickler, Axel Wüstemann:</i> NetWare High technology with PC networks	46
<i>Dieter Lenz, Volker Nawrotzki, Martin Holzhauser:</i> HODIS-2 room reservation system based on SCOM-LAN	50

Verlag Die Wirtschaft Berlin
Am Friedrichshain 22, Berlin 1055
Verlagsdirektor: Dieter Grüneberg

edv-aspekte
Zeitschrift für spezielle Themen
der Informationsverarbeitung,
herausgegeben von der Redaktion
rechenstechnik/datenverarbeitung,
1055 Berlin, Am Friedrichshain 22
Chefredakteur: Franz Loll 4 38 73 41
Redakteurin: Claudia Schulz 4 38 73 16
Sekretariat: 4 38 72 33
Fernschreiber: 114 566
Gestaltung: Marlies Hawemann
Redaktionschluss: 27. 2. 1989

Lizenz des Presseamtes beim Vorsitzenden
des Ministerrates der DDR Nr. 1529

edv-aspekte
Erscheinungsweise vierteljährlich zum Bezugs-
preis DDR 5,00 M je Heft
EDV-Artikel-Nr. 1331
Auslandpreise sind dem Zeitschriften-
katalog des Außenhandelsbetriebes
Buchexport zu entnehmen.

Satz: Verlag Die Wirtschaft, Berlin
Druck: (140) „Neues Deutschland“, Berlin

Anzeigenverwaltung:
Berliner Verlag,
Karl-Liebknecht-Str. 29
Berlin 1056 Telefon: 2 70 33 02

Anzeigenannahme:
Berliner Verlag und Annahmestellen
in Berlin und in den Bezirken
Zur Zeit gültige Anzeigenpreisliste Nr. 12

Im Ausland:
INTERWERBUNG GmbH – Gesellschaft
Werbung und Auslandsmessen der DDR,
Mann-Duncker-Str. 89 Berlin 1157

Bestellungen aus der DDR sind an den
Postzeitungsvertrieb zu richten,
Inkasso-Zeitraum: vierteljährlich

Im Ausland:
In den sozialistischen Ländern nur der zustän-
dige Postzeitungsvertrieb. In allen anderen
Staaten der örtliche Buch- und Zeitschriften-
handel. Bestellungen des Buch- und Zeit-
schriftenhandels sind zu richten an

BUCHEXPORT
Volkseigener Außenhandelsbetrieb der DDR,
DDR – Leninstr. 16, Leipzig 7010,
Postfach 160
oder an Verlag Die Wirtschaft,
DDR – Am Friedrichshain 22, Berlin 1055

Mitglieder des Redaktionsbeirates
Dr. Claus Goedecke · Dr. Rolf Gräßler
Prof. Dr. sc. Gerhard Keßler · Dr. Rolf Killian
Hans Kunau · Walter Münch · Axel Rath sack
Prof. Dr. sc. Gerd Rossa ·
Prof. Dr. sc. Claus Sattler ·
Prof. Dr. sc. Wolfgang Schoppa (Vorsitzender)
Dr. Werner Schulze · Horst Stoll
Prof. Dr. Franz Stuchlik · Dr. Dieter Urban

Verteilte Verarbeitung

In den letzten Jahren hat die dezentrale Verarbeitung von Daten und Informationen eine sehr beachtliche Entwicklung durchlaufen. Durch die Dezentralisierung der Rechenleistungen sind die informationsverarbeitenden Prozesse wieder den funktionellen betrieblichen Strukturen angegliedert worden.

Der hohe materielle Aufwand, der sich hinter dem breiten Einsatz der Schlüsseltechnologien, insbesondere den Personalcomputern, verbirgt, muß sich jedoch noch schneller in ökonomische Effektivität umsetzen. Eine weitergehende Nutzungsart ist in der Rechnerkommunikation zu sehen.

Insbesondere der Zugriff auf entfernte Datenbestände eröffnet die Möglichkeit der arbeitsteiligen Informationsverarbeitung. Alle bisher nur theoretisch gültigen Regeln der Datenverarbeitung können endlich realisiert werden:

- jedes Datum soll nur einmal erfaßt werden,
- jedes Datum soll möglichst dort erfaßt werden, wo es existiert,
- jedes Datum soll dort verwaltet werden, wo es am häufigsten benötigt wird usw.

Diese, uns allen seit langem bekannten Regeln, konnten weder in der Zeit der zentralen Datenverarbeitung in den Organisations- und Rechenzentren noch durch die dezentralen PC realisiert werden, obwohl jeder in der DV-Praxis tätige Mitarbeiter sie unbestritten befürwortete. Die verteilte Verarbeitung macht es nun endlich möglich. Aber wie jeder qualitative Entwicklungssprung setzt sich auch dieser nicht im Selbstlauf durch!

Eine „Einsatzvorbereitung“ der verteilten Verarbeitung hat mit Sicherheit einen weit höheren Grad an Komplexität zu berücksichtigen als in den bisher üblichen Fällen. Diese Komplexität stellt sich in mehreren Dimensionen und Ebenen dar.

Der Einsatz des ersten Personalcomputers impliziert bereits die Problematik der Rechnerkommunikation, auch wenn man zu dem Zeitpunkt noch nicht daran denkt. Das heißt Einsatzvorbereitung von LAN und PC können einander bedingen und überdecken.

Obwohl theoretisch relativ uninteressant, stellt sich z. B. die Frage der Verkabelung im Betrieb sehr oft als ein Problem heraus. Umso mehr ein Grund, ihr die entsprechende Aufmerksamkeit zu widmen. Ein Leitungs- und Entscheidungsaspekt sollte deshalb sein, daß bei jedem Neubau und jeder größeren Werterhaltungsmaßnahme neben der Frage Telefon und E-Versorgung auch die Informationsübertragung beachtet wird. Dieser Teil erfordert genauso sein abgestimmtes Projekt wie Telefon, Raumschutz, Klima, Wechselsprechanlagen usw. Die für den Komplex ODV (Organisation und Datenverarbeitung) verantwortlichen Leiter haben hier eine besondere Verpflichtung.

Mit dem Netzaufbau ist jedoch erst die Basis geschaffen. Ein Mailsystem oder ein Filetransfer ist noch lange keine verteilte Verarbeitung. DV-typische Probleme ergeben sich aus der isolierten Entwicklung der Einzellösungen, die Unverträglichkeiten der Datenformate und der Schlüsselssysteme bedingen, und andere Aspekte der Datenrepräsentation betreffen. Womit dann auch der Bogen bis hin zu den Fragen der Betriebsorganisation gespannt ist. Es liegt mehr als ein Wortspiel darin, wenn man sagt:

„Ist die Verarbeitung verteilt, muß die Organisation zentralisiert sein.“

Wobei die Einheitlichkeit z. B. der Primärdatenorganisation noch ein zumindest theoretisch überschaubares Problem darstellt. In der Praxis können sich jedoch oft unüberwindliche Probleme ergeben. Notwendige Voraussetzungen für die Funktion verteilter Systeme ist aber diese Einheitlichkeit, hinreichende Bedingung für ihre effektive Arbeit ist jedoch eine ihrer Arbeitsweise adäquate neue Struktur der funktionellen Prozesse.

Diese neuen Strukturformen bewirken Delegationen von Kompetenzen, Verantwortlichkeiten sowie Arbeitsinhalten und Arbeitsformen. Schließlich und endlich muß klar sein, was man verteilen bzw. verteilt verarbeiten will und wer in dem verteilten System welche Befugnisse hat (Zugriffsschutz).

Es können Funktionen, Daten und Belastungen das Hauptziel eines Systementwurfes sein. Im Wechselspiel dazu bedingen die Funktionen die Anforderungen an die Kommunikation als Basis jeder verteilten Verarbeitung. An dieser Stelle ist mit Sicherheit die Frage der Fehlertoleranz zu diskutieren.

Der Einsatz in der Fertigung unterscheidet sich nicht nur technisch-physikalisch von einer Anwendung in Büro oder Verwaltung oder gar in Konstruktion und Entwicklung. Neu durchdacht werden müssen in diesem Zusammenhang auch die Methoden und Konzepte der Datenbanktechnologie.

Die Bereiche der Datenbankverwaltung haben neue Anwendungen, Einsatzgebiete, zahlreiche neuartige Forschungsprobleme und herausfordernde Entwicklungs-Aufgaben hervorgebracht. Die vorliegende aspekte-Ausgabe versucht, zu einigen dieser Problemkreise sowohl aus der Sicht der Forschung als auch aus der des Anwenders einige Anregungen zu geben.

Prof. Dr. sc. Gerd Rossa

Rechnergestützte Betriebswirtschaft und verteilte Verarbeitung

Prof. Dr. Gerd Rossa, Wilhelm-Pieck-Universität Rostock
Institut für sozialistische Wirtschaftsführung, Informatik-Labor

Im Zuge des umfassenden Einsatzes von Informations- und Kommunikationstechnik in unseren Betrieben und Kombinat ergibt sich die Möglichkeit, dezentral vorhandenes aktuelles Datenmaterial einer zusätzlichen Informationsverarbeitung nutzbar zu machen. Dieses Datenmaterial repräsentiert, obwohl oft isoliert entstanden und dezentral verwaltet, den betrieblichen Reproduktionsprozeß als Ganzes.

Damit ergeben sich neben den bereits zentral realisierten Berichtserstattungsfunktionen neue methodische Möglichkeiten für die sozialistische Betriebswirtschaft (SBW).

Andererseits ist die SBW aufgrund der veränderten Dynamik des Reproduktionsprozesses gezwungen, sich neue, vor allem zeiteffektive und analytisch wirksame Methoden zu erschließen.

Die Computerisierung der SBW läßt sich aus heutiger Sicht in folgende Etappen gliedern:

1. rechnerunterstützte Einplatz-Systeme
2. rechnergestützte Einplatz-Systeme
3. hierarchisch datenintegrierte Einplatz-Systeme
4. horizontal datenintegrierte rechnergestützte Systeme
5. CIM-Konzepte.

Der qualitative Unterschied zwischen rechnerunterstützt und rechnergestützt besteht darin, daß bei der Rechnerunterstützung vorhandene konventionelle Funktionen im Sinne einer einfachen Rationalisierung automatisiert werden, ohne daß sich die Funktion in wesentlichen Zügen ändert. Als Beispiel kann hierfür der computerunterstützte Fahrkartenverkauf der DR angesehen werden. Aus demselben Bereich kann gleichzeitig das rechnergestützte System der Platzbuchung, das neue Eigenschaften hat (wie z. B. die Platzbuchung in anderen Städten), die nur auf der Basis der Rechnerkommunikation möglich sind, als Beispiel für rechnergestützte Systeme dienen.

Der Übergang von der zweiten zur dritten Stufe ergibt mit der Datenintegration über mehrere Teilsysteme wiederum die Möglichkeit, neue Methoden zu erschließen.

Eine vollständige Datenintegration, ein-

schließlich der Umgebungsdaten, bildet die Basis aller CIM-Konzepte.

Entsprechend der obigen Gliederung sind auch die unterschiedlichen Ansätze zur konzeptionellen Herausbildung einer rechnergestützten Betriebswirtschaft einzuordnen:

- Top-Down-Entwurf
ausgehend von der Zielvorstellung eines Gesamtsystems nach CIM-Konzepten

- Bottom-Up-Entwurf
Analyse und Erweiterung der bestehenden Elemente einer rechnergestützten Betriebswirtschaft (RBW).

Beide Konzepte sind in ihren reinen Formen nicht tragfähig. Sinnvollerweise sollte, von einer relativ groben Zielstellung ausgehend, das System der betrieblichen DV analysiert und schrittweise erweitert werden. Im Verlaufe dieses Prozesses wird die Zielvorstellung permanent zu aktualisieren sein.

Die RBW hat zur Realisierung dieser Stufen unter Beachtung der existierenden Ausgangssituation deshalb zuerst analytische und methodische Fragen bzw. Probleme zu behandeln. Beispielfolgend sollen einige aufgezählt werden:

- Welche Aufgaben ergeben sich aus der zu akzeptierenden Existenz eines Systems verteilter Datenbestände aus der Sicht der SBW für die Informatik?

- Welche qualitativ neuen Möglichkeiten ergeben sich für die SBW durch den Computereinsatz in einzelnen Bereichen? Zum Beispiel Bestellpunktermittlung in der Materialwirtschaft (Logistik)

- Welche neuen Möglichkeiten ergeben sich für die SBW durch eine Kommunikation zwischen einzelnen Teilsystemen (horizontal)? Zum Beispiel Kostenanalysen bereits im CAD-Bereich

- Welche Mittel und Methoden lassen sich einsetzen, wenn eine hierarchische Datenintegration möglich ist? Zum Beispiel kann dann der Bereich Ökonomie weit besser das Informationsloch zwischen Plan und Abrechnung ausfüllen und aus ökonomischer Sicht auf die Produktionsplanung direkten Einfluß nehmen

- Welche neuen betriebsorganisatorischen und Leitungsstrukturen ergeben sich im Prozeß des Aufbaues eines sol-

chen verteilten aber dennoch integrierten Systems der RBW?

Die Rechnerkommunikation als Voraussetzung der rechnergestützten Betriebswirtschaft

Ziel aller Konzepte des Einsatzes von Computern in betrieblichen Bereichen ist es, den Produktionsprozeß zu beschleunigen, die Produktionskosten zu senken, die Flexibilität zu verbessern und gleichzeitig die Produktqualität zu erhöhen.

Um die dafür notwendigen schnelleren Informations- und Materialflußkanäle bereitzustellen, sind neben organisatorischen Gesichtspunkten zwei Schlüsseltechnologien aus dem Bereich der Informatik von grundsätzlicher Bedeutung. Es sind dies die Datenbanken und die Computerkommunikation auf der Basis von Netzwerken.

Für den Datentransport zwischen den einzelnen betrieblichen Bereichen bzw. innerhalb der Bereiche wird zukünftig der Einsatz lokaler Netze beträchtlich zunehmen. Diese bilden das Rückgrat und die Voraussetzung für die Systemintegration zu einem gleichberechtigten Rechnernetz, bestehend aus Großrechnern, Arbeitsstationen, Personalcomputern und rechnergestützten Steuerungen.

Die bezüglich der Datenverteilung unterschiedlichen Hierarchieebenen im Betrieb führen gewöhnlich zu entsprechenden Rechnerhierarchien.

Auf der Produktions-Planungs-Ebene sind gewöhnlich nur wenige, dafür aber leistungsstarke ESER-Rechner mit großen Speicherkapazitäten und Rechenleistungen installiert. Charakteristisch sind eine lange Datenlebensdauer und umfangreiche Datenmengen. Datenbewegungen auf dieser Ebene sind relativ selten.

Umgekehrt lassen sich auf der Ebene der Prozeßperipherie nahezu gegensätzliche Aussagen zu den erwähnten Merkmalen finden. Hier liegt eine Vielzahl von Komponenten vor, angefangen von Stellgeräten, Feldreglern, Wertgebern, Antrieben und Sensoren bis hin zu einfachen speicher-programmierbaren Steuerungen. Alle diese peripheren Ge-

räte erzeugen hinsichtlich ihrer Existenzdauer extrem kurzlebige Daten, die nur wenige Byte umfassen und an übergeordnete Stellen oder Ebenen weitergegeben werden.

Die Kommunikationsrichtung ist ausschließlich vertikal. In bezug auf die Verarbeitung und die Speicherung dieser Daten sind Echtzeiteigenschaften gefordert. Die lokalen Rechen- und Speicherkapazitäten können als relativ gering eingestuft werden.

Diese zwischen unterster und oberster Ebene festzustellenden gegensätzlichen Eigenheiten stehen, über die dazwischen liegenden Ebenen gesehen, in einem fast linearen Zusammenhang. Das informations-technische Werkzeug zur Integration der verschiedenen Rechnersysteme innerhalb und über die Hierarchieebenen hinweg, bilden die lokalen Netze (LAN).

Eine weitere Schlußfolgerung aus der bisherigen Entwicklung ist, daß ein einziger Typ eines lokalen Netzes mit Sicherheit nicht ausreichen wird, um den unterschiedlichen Anforderungen an die Übertragung von Daten gerecht zu werden.

Die Verbindung heterogener LAN kann mit den LAN-Architektur-Komponenten Repeater (Verstärker), Bridge (Brücke), Route (automatische Wegwahl) und Gateway (Protokollumsetzer) realisiert werden. Die entscheidende Schwierigkeit, die in diesem Zusammenhang überwunden werden muß, liegt in der Inkompatibilität der vorzufindenden Rechnersysteme und ihrer LAN-Schnittstellen begründet.

Die integrierte betriebliche Datenhaltung als Voraussetzung der rechnergestützten Betriebswirtschaft

Sehr eng an die Netzwerke sind dezentrale oder verteilte Datenbanksysteme gebunden. Der Schwerpunkt wird in nächster Zukunft auf homogen verteilten Datenbanksystemen liegen. Also auf Systemen, deren lokale Datenbanken untereinander kompatibel sind und die auf verschiedenen Rechnerklassen implementiert werden.

Der Datenzugriff auf Fremdsysteme

ist oft nur über spezielle Schnittstellen-Programme möglich, die wegen der Umsetzung der verschiedenen Daten-Struktur-Formate die Zugriffszeit noch weiterhin negativ beeinflussen können. Ein derartiger übergreifender, d. h. transparenter Datenzugriff ist aber eine notwendige Voraussetzung der RBW. Ein Ansatz wird in der Arbeit von Heidrun Ortleb in diesem Heft angedeutet. Die Spezifikation von Datenstrukturen, der effiziente Zugriff darauf, ihre leichte Manipulation und konsistente, möglichst redundanzfreie Verwaltung bedürfen deshalb bei jeder größeren Datenverarbeitungsanwendung besonderes Augenmerk. Konzeptuelle Fehler auf dieser Ebene führen in fast allen Fällen zu einem kostenaufwendigen Überarbeiten des Gesamtsystems.

Da zum Beispiel einer RBW-Realisierung oder CAD/CAM-Systemen weitreichende Investitionsentscheidungen in bedeutender Höhe vorausgehen, muß in jedem Bereich, der eine solche Lösung anstrebt, die Aufteilung und Haltung der gesamten betrieblichen Datenbasis besonders beachtet werden.

Wesentlich für die Betrachtung dieser Daten sind der Ort ihrer Entstehung, ihre interne Strukturierung und ihre mehrfache Verwendung in den verschiedenen betrieblichen Bereichen.

Dieser Aspekt muß Grundlage jeder Entwicklung im Bereich der RBW sein. Alle in einem Modell der RBW benötigten Daten sind mit großer Sicherheit bereits irgendwo maschinenlesbar vorhanden.

Im Produktions-Planungs-Bereich sind das Stamm-, Auftrags-, Kunden-, Termin-, Material- und Kapazitätsdaten. Der CAD-Bereich erzeugt dagegen vornehmlich Geometrie- und Stücklisten-daten.

Im CAP-Bereich fallen Technologie-, Arbeits- und Fertigungsplandaten an. In der Arbeitsplanung besteht ebenfalls die Forderung nach einer Mehrfachverwendung der im Konstruktionsbereich erstellten Geometriedaten, um sie bei der Arbeitsplanerstellung um Technologiedaten zu ergänzen.

In der Konstruktion können kostenwirksamere Entscheidungen gefällt wer-

den, wenn verdichtete Kalkulationsdaten aus dem PPS-Bereich verfügbar sind. Diese Beschreibung der Lokalitäten und der Verarbeitung gleicher Daten in verschiedenen Bereichen eines Betriebes macht es naheliegend, sie in einer logischen Datenbasis zu speichern.

Andererseits wird mit der Dezentralisierung der Automatisierungssysteme eine eindeutige Aufgabenzuordnung auf die einzelnen, in den verschiedenen Betriebsbereichen verteilten Systeme notwendig. Bei fast allen Automatisierungsaufgaben führt die Zuweisung von Funktionen zu einer hierarchischen Struktur des Datenverarbeitungssystems. Für einen Betrieb lassen sich üblicherweise vier Ebenen in der Hierarchie unterscheiden. Das sind, von der Gesamtleitung angefangen,

1. Leitungssystem mit der Produktionsplanung, darunter die

2. Fertigungs- und Produktionsleitungsebene, die wiederum der

3. Prozeßführungsebene übergeordnet ist, die ihre Steuerdaten an die Steuerung und Regelung in den einzelnen Bereichen übergibt.

Ganz unten ist der Bereich der

4. Betriebsdatenerfassung, der Meßgeber und der gesteuerten Antriebe einzuordnen.

Der Vorteil einer solchen allgemeinen Strukturierung liegt darin, daß jede Ebene überwiegend mit den Daten arbeitet, die ihr zugeordnet sind. Z. B. sind die logistischen Daten für die Koordination des Gesamtfertigungsablaufes in der Produktionsleitungsebene angesiedelt.

Die auf den einzelnen Ebenen anfallenden Daten werden heute auf zwei prinzipiell unterschiedliche Arten verwaltet, in Dateiverwaltungssystemen oder Datenbankverwaltungssystemen. Dabei ist zu bemerken, daß der Trend eindeutig in Richtung der Datenbanksysteme geht. Bezüglich der Verteilung läßt sich die Datenorganisation in drei Formen unterscheiden:

- dezentrale Datenhaltung
- zentrale Datenhaltung (eventuell mit Fernzugriff)
- verteilte Datenhaltung.

Die dezentrale oder lokale Datenhaltung stellt eine besonders in der kommerziellen Datenverarbeitung bewährte Lösung mit einem schnellen Zugriff auf den verfügbaren lokalen Datenbestand dar. Sie bildet das Einstiegsniveau der dezentralen Datenverarbeitung. Um aber mit anderen betrieblichen Bereichen Daten auszutauschen oder auf diese zugreifen zu können, sind unhandliche Datentransferprogramme zwischen den beteiligten Datenverwaltungssystemen notwendig. Weiterhin ergibt sich aus einer solchen Konzeption ein hoher Aufwand zur Gewährleistung der Konsistenz des Gesamtdatenbestandes.

Bei einer zentralen Datenhaltung, wie sie aus der zentralen Rechentechnik im ORZ übernommen wurde (wenn auch mit Fernzugriff), ist das Problem der Datenkonsistenz zwar mit Sicherheit gelöst, der Datenzugriff dafür aber langsamer und relativ starr.

Da der Datenbestand für alle Anwender gleichrangig zentral gehalten wird, birgt der Dateiserver, der in diesem Fall die Basis einer zentralen Datenhaltung darstellt, die Gefahr eines Engpasses in sich.

Die dritte und neueste Organisationsform, die verteilte Datenhaltung, stellt eine aufwendigere Lösung dar, sie erlaubt aber einen schnellen und bequemen Datenzugriff auf lokale Bestände und garantiert gleichzeitig die Konsistenz der verwalteten Daten. Der Datenbestand erscheint für den Anwender transparent. Das heißt der gesamte Datensatz ist für ihn wie ein lokaler Bestand und erfüllt damit die vorher hergeleiteten Anforderungen einer logisch zentralen Datensicht bei verteilten physikalischen Speicherstrukturen. Die Dezentralität der Datenhaltung bedingt andererseits jedoch einen hohen Kommunikations- und Koordinierungsaufwand, der sich vor allem bei menuegesteuerten Datenbankanwendungen in einem verschlechterten Antwort-Zeit-Verhalten niederschlagen kann. Ein weiteres Problem stellt ein eventueller Datenverlust dar. Dagegen muß durch ausreichende Sicherungsstrategien Vorsorge getroffen werden. Die Anforderungen

an Datenbank-Management-Systeme auf den unterschiedlichen Ebenen sind zum Teil sehr gegensätzlich. Klassische Systeme, wie sie auf den oberen Leitungsebenen seit längerer Zeit erfolgreich im Einsatz in der Verwaltung sind, lassen sich mit der ihnen unterlegten Struktur schlecht auf andere Bereiche, wie Konstruktion oder PPS übertragen.

Den Zugriff und die Manipulation komplexer Objekte, wie sie z. B. Zeichnungen im Konstruktionsbüro sind, erlauben heutige Datenbanksysteme in der Regel überhaupt nicht. Außerdem müssen auf der Prozeßführungs-Steuerungs-Regelungs-Ebene neben teilweise Echtzeitbedingungen besonders auch die verteilte physikalische Bereitstellung der benötigten Daten berücksichtigt werden. Aus diesen Gegebenheiten läßt sich der Schluß ziehen, daß es das alle Wünsche befriedigende, universelle Datenhaltungssystem für den praktischen Einsatz sicherlich nicht geben wird, falls nicht völlig neue Modelle der Datenverwaltung unter dem Aspekt der Verteiltheit der Daten entwickelt werden.

Aus der Sicht der Nutzung der Daten sind die Entscheidungsunterstützungssysteme (DSS) oder auch die Methodenbanksysteme (MBS) eine wichtige Komponente der rechnergestützten Betriebswirtschaft. Sie bieten eine Hilfe bei der Lösung schlecht strukturierter Probleme an. Sie gehen damit über das Leistungsprofil der Informationssysteme hinaus und nutzen dazu moderne softwaretechnologische Prinzipien wie wissensbasierte Systeme u. ä. DSS bestehen aus folgenden Komponenten:

- Sprach-System
- Datenbasis
- Wissens-System
- Problemverarbeitungs-System.

Das Steuersystem – der DSS-Monitor – analysiert die Aufgabe und gibt Nachrichten an die Basisprozessoren (Anfragen oder Ergebnisse von Anfragen) weiter. Der DSS-Monitor ist selbst ein wissensbasiertes System, das Strategien in Form von Regeln und Fakten erarbeitet und nutzt.

Mit diesem kurzen Hinweis auf die DSS bzw. MBS sollte deutlich gemacht wer-

den, daß für die RBW nicht nur die Datenbasis eine große Bedeutung hat, sondern daß die Algorithmen der Verarbeitung dieser Daten eine gleichrangige (wenn nicht noch höhere) Bedeutung haben.

Die Algorithmen sind aber weniger ein Problem der Informatik als der Betriebswirtschaft selbst.

→ Veranstaltung

Wechselbeziehungen zwischen Informationssystem, Arbeitsgestaltung und Organisationsgestaltung

Termin: 10.–13. Juli 1989

Ort: Berlin

Veranstalter: Humboldt-Universität zu Berlin, Sektion Wissenschaftstheorie und -organisation, Bereich Systemgestaltung und automatisierte Informationsverarbeitung; IFIP, TC 9 Computer and Society, Working Group 9.1, Computer and Work; Kammer der Technik (WGMA); GI der DDR

Schwerpunkte:

- Herausforderungen moderner Informations- und Kommunikationstechnologien an die Informationssystem-, Arbeits- und Organisationsgestaltung
- Implikationen der Organisationsstruktur und Verhalten
- Die Implikationen der Mikrocomputer, der Mensch-Maschine-Interaktion
- Einbeziehung menschlicher Werte in den Gestaltungsprozeß
- Möglichkeiten der Demokratisierung sozialer Organisationen
- Bewertung alternativer Strategien der Informationsgestaltung.

Organisatorische Hinweise:

Interessenten wenden sich bitte an die Humboldt-Universität zu Berlin, Sektion WTO, Prof. Dr. K. Fuchs-Kittowski, PF 1297, August-Bebel-Platz / Kommode, Berlin, 1086.

Gestaltung netzfähiger Software für Arbeitsprozesse im Büro

Prof. Dr. Claus Sattler

Institut für Informatik und Rechentechnik der AdW der DDR

Mit der Bereitstellung von lokalen und Weitverkehrsnetzen ergeben sich Möglichkeiten der Verteilung von Funktionen zur Unterstützung von Arbeitsprozessen im Büro auf verschiedene in ein Netz integrierte Rechner. Eine solche Verteilung kann Kostenvorteile bringen, wenn z. B. eine teure Ressource über das Netz mehreren Nutzern zugänglich gemacht wird. Sie kann aus administrativen oder funktionellen Erwägungen erfolgen, wie z. B. im Falle eines elektronischen Mitteilungssystems. Der Entwurf durchgängiger Systeme verlangt eine einheitliche Gestaltung der Software der einzelnen Komponenten. Diesem Anliegen dient der Entwurf eines ISO-Standards „Modell verteilter Büroanwendungen“ /1/, der im weiteren beschrieben werden soll.

Mit der Erarbeitung von Standards zur Verkopplung von Rechnern über lokale und Weitverkehrsnetze hat die Internationale Standardisierungsorganisation ISO, deren Mitglied die DDR seit 1988 ist, Grundlagen für den Aufbau von Systemen der verteilten Verarbeitung geschaffen. Letztlich besteht das Ziel der auf der Basis des Referenzmodells zur Verkopplung offener Systeme /2/ erfolgten Spezifikation der Dienste und Protokolle der 7 Schichten in der Schaffung einheitlicher Prinzipien für die Kommunikation zwischen Anwendungsprozessen, die auf unterschiedlichen Rechnern ablaufen.

In letzter Zeit wendet sich die ISO mehr und mehr auch den verteilten Anwendungen selbst zu. Beispiele sind die Arbeiten zur Standardisierung zum Mitteilungsorientierten Textaustausch (MOTIS, in Abstimmung mit den CCITT-Empfehlungen X.400ff.) und zum Verzeichnis (Directory, in Abstimmung mit den CCITT-Empfehlungen X.500). Dabei hat sich sehr schnell die Erkenntnis eingestellt, daß diese verteilten Anwendungen nicht isoliert betrachtet werden können, sondern aus der Sicht des Nutzers miteinander in Wechselwirkung stehen und deshalb nach einheitlichen Gesichtspunkten zu gestalten sind. Das betrifft solche Aspekte wie die Architektur (Struktur, Operationen), die Adressierung, Schutzfragen und weitere.

Ergebnis dieses Erkenntnisprozesses ist der Entwurf des ISO-Standards DP10031 „Distributes-Office-Applications Model“ /1/, der im Teil 1 die allgemeine Modellbeschreibung und im Teil 2 die Beschreibung des sog. Referenced Data Transfer enthält. Eine verteilte (Büro-)Anwendung wird darin definiert als *Satz von über ein oder mehrere offene Systeme verteilten Informationsverarbeitungsressourcen, der dem (menschlichen) Nutzer eine definierte Funktionalität zur Lösung einer gegebenen (Büro-)Aufgabe bereitstellt*. Anforderungen der Nutzer an ein verteiltes Bürosystem, denen das Modell entsprechen soll, sind in Tab. 1 zusammengefaßt. Diese Liste ist offen für zukünftige weitere Anwendungen. Die Nutzung dieser Anwendungen ist gekennzeichnet durch einen hohen Integrationsgrad, z. B. kann ein Nutzer ein Dokument aus seinem elektronischen Briefkasten auswählen, in der elektronischen Ablage ablegen und auf einem Druckerserver drucken lassen. Ein anderes Beispiel ist das Versenden eines Dokuments, wo zunächst über das Verzeichnis AdreBinformationen zu beschaffen sind und das Dokument mit einem Absende-Zeitstempel zu versehen ist.

Die Anwendungen lassen sich zwei Klassen zuordnen:

- Produktive Anwendungen sind sichtbar für den menschlichen Nutzer und werden durch ihn genutzt. Zu ihnen gehören entfernte Drucker, Dokumentenablage und -wiederauffinden, elektronische Post, bestimmte Nutzungsformen des Verzeichnisses.

- Unterstützende Anwendungen bilden die Netzbetriebsumgebung für die produktiven Anwendungen und deren Nutzer. Zu ihnen gehören die Komponente für die Bezugszeit, die Sicherheit unterstützenden Anwendungen, bestimmte Nutzungsformen des Verzeichnisses und die Unterstützung des Datentransfers zwischen verschiedenen Anwendungen.

Basiskonzepte

Eine verteilte Büroanwendung besteht aus Knoten, die durch ein Netz mitein-

- Interpersoneller Mitteilungsaustausch, um mit anderen Nutzern zu kommunizieren,
- Gruppenkommunikation, um mit Nutzergruppen zu kommunizieren,
- Konvertierung, um den Austausch von Dokumenten mit unterschiedlichen Darstellungsprotokollen oder unterschiedlicher Zeichenkodierung zu ermöglichen,
- Abspeichern und Wiederauffinden von Dokumenten, um ein geordnetes Abspeichern und Wiederauffinden nach Mehrfachschlüsseln zu ermöglichen,
- Ein- und Ausgabe von Dokumenten über unterschiedliche E/A-Geräte, z. B. Scanner, Drucker,
- Verzeichnis, um zu erfahren, wo und wie der Zugriff zu entfernten Kommunikationselementen, Anwendungen oder Nutzern zu erfolgen hat,
- Authentizität, um nichtautorisierten Zugriff zu den verschiedenen Anwendungen zu verhindern,
- Bezugszeit, auf die lokal zugegriffen werden kann, für solche Zwecke wie das „Zeitstempeln“ von Mitteilungen und Files im Netz,
- Direktzugriff zu entfernten Servern (z. B. Videotex) und Nutzern (z. B. über Teletex),
- Indirekte Kommunikation (Store and forward) mit entfernten Systemen, um nicht zeitkritische Informationen zu übertragen,
- Datenaustausch zwischen unterschiedlichen Anwendungen oder entfernten Servern.

Tab. 1 Anforderungen von Nutzern an ein verteiltes Bürosystem

ander verbunden sind. Es werden unterschieden:

- Nutzerknoten und
- Serverknoten.

Ein Nutzerknoten ist ein Gerät, mit dem ein Nutzer direkt in Beziehung steht und der direkte interaktive Funktionen bereitstellt.

Ein Serverknoten ist ein Gerät, das Ressourcen verwaltet, die zwischen vielen Nutzern geteilt werden.

Eine Interaktion eines Nutzers mit einem verteilten Bürosystem kann eine Anzahl von Aktivitäten auf einer Anzahl von Knoten hervorrufen.

Die Interaktionen zwischen einer Aktivität auf einem Knoten und einer Aktivität auf einem anderen Knoten werden

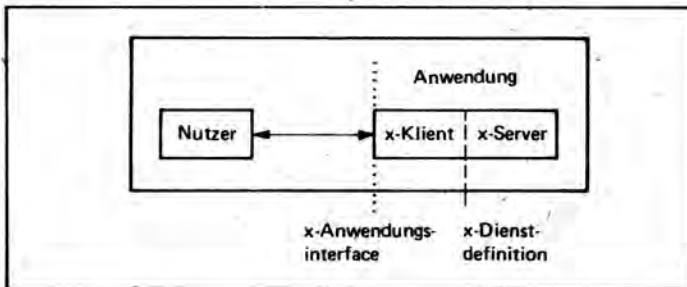


Abb. 1a Nicht-verteilte Büroanwendung

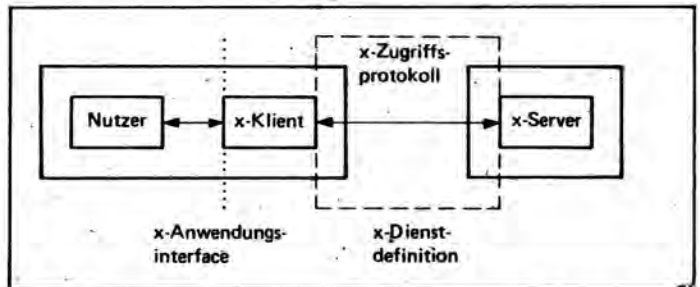


Abb. 1b Verteilte Büroanwendung

(leicht vereinfacht) im Referenzmodell zur Verkopplung offener Systeme durch Interaktionen zwischen einem Paar von Anwendungsprozessen, von dem sich auf jedem Knoten einer befindet, repräsentiert.

Klient-Server-Modell

Wesentliches Basiskonzept für die Modellierung verteilter Büroanwendungen ist das Klient-Server-Modell. Eine Anwendung x stellt sich einem Nutzer über das x -Anwendungs-Interface dar, das beliebig ausgestaltet sein kann und keiner Standardisierung unterliegt. Ist diese x -Anwendung für eine Verteilung auf mehrere Knoten vorgesehen, so wird eine x -Dienst-Definition als Ausgangspunkt für eine Verteilung benötigt. Die x -Anwendung unterteilt sich dazu in den x -Klient, der das x -Anwendungs-Interface auf die x -Dienst-Definition abbildet, und den x -Server, der den Dienst erbringt (Abb. 1a).

Die Verteilung einer x -Anwendung (Abb. 1b) sollte für den Nutzer transparent sein, d. h. das x -Anwendungs-Interface soll erhalten bleiben. Der x -Server

rückt an einen anderen Ort. Dann kommunizieren x -Klient und x -Server über ein x -Zugriffs-Protokoll. Die x -Dienst-Definition schließt das x -Zugriffs-Protokoll ein. Beide müssen standardisiert werden.

Abb. 2a stellt in Erweiterung von Abb. 1a eine nicht-verteilte Büroanwendung dar, bei der ein Nutzer die Anwendungen x und y nutzt (z. B. x = Lesen eines Files, y = Drucken eines Files). Die Komponente „Nutzer“ integriert beide Anwendungen (das gelesene File wird an den Drucker übergeben). Abb. 2b zeigt die gleiche Anwendung bei Nutzung entfernter Server (x = File-Server, y = Drucker-Server). Der Nutzer-Anwendungs-Prozess charakterisiert die Struktur eines multifunktionalen Nutzersystems.

System aus Servern des gleichen Typs

In einem zweiten Verteilungsschritt läßt sich die Funktionalität eines Servers einer Anwendung auf mehrere Server auf verschiedenen Knoten verteilen. Die Server einer x -Anwendung bilden dann das x -System. Die Server sind funktionell

insofern gleich, daß jeder das gleiche Zugriffsprotokoll unterstützt. Jeder Server befindet sich in einem Knoten. Ein x -System kann somit bestehen aus (Abb. 3)

- einem einfachen Server
- mehreren nicht-interagierenden Servern

Beispiel ist ein Filesystem, dessen Files auf mehreren Fileservern gespeichert sind und in dem der Nutzer wissen muß, auf welchem Server sich das benötigte File befindet.

- mehreren interagierenden Servern, die in diesem Fall über ein Anwendungs-System-Protokoll kooperieren.
- Beispiel ist eine verteilte Datenbank, bei der der Nutzer sich immer an den gleichen Server wendet und dort nicht verfügbare Informationen innerhalb der verteilten Datenbank beschafft werden.

Wechselwirkung zwischen Servern verschiedener Typs

Für die Wechselwirkung zwischen Servern verschiedener Anwendungen enthält das Modell zwei Formen:

Abb. 2a Multiple nicht-verteilte Büroanwendung

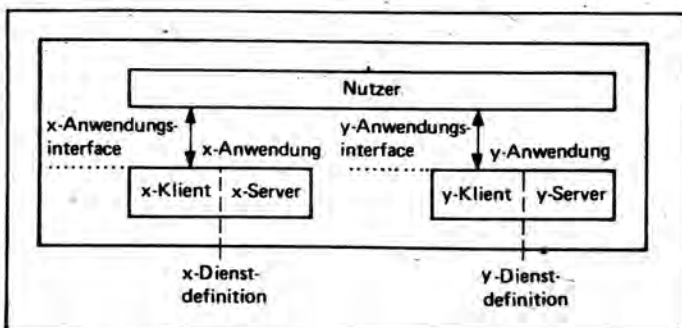
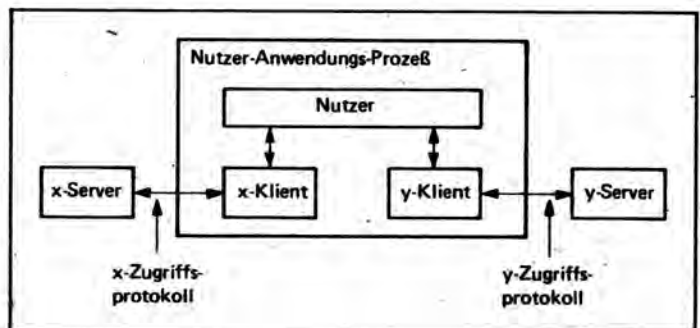
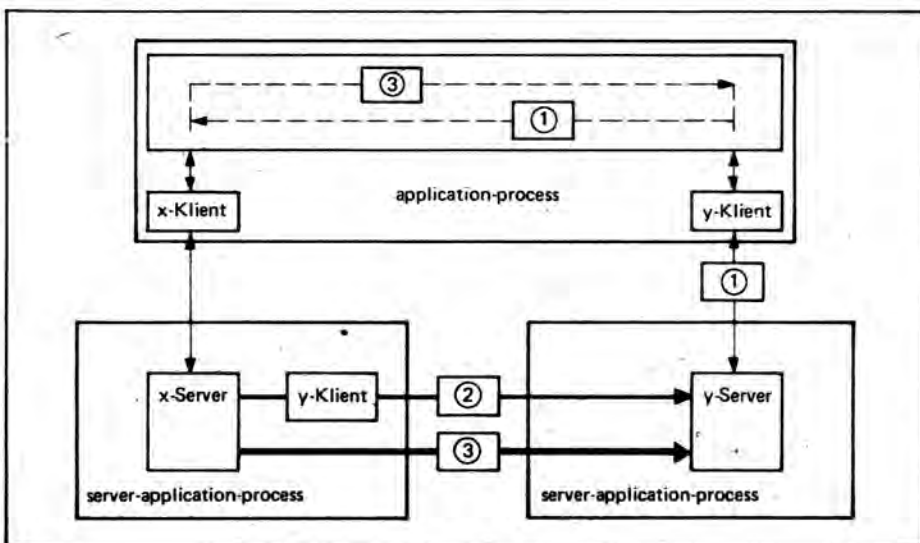
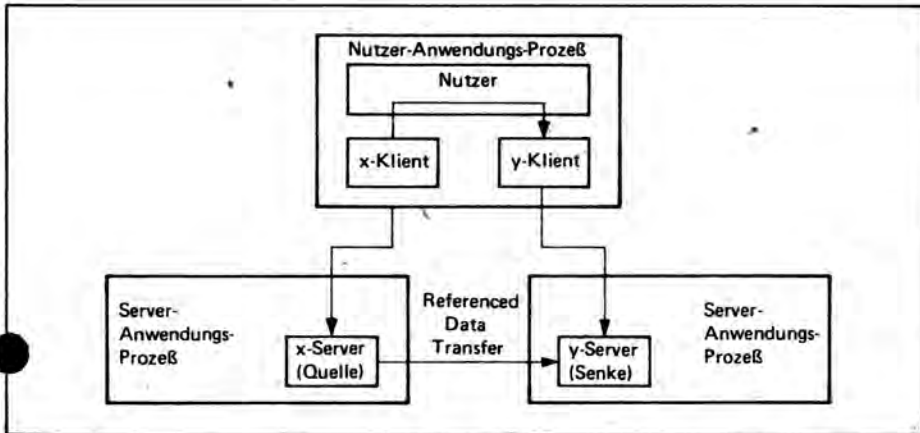
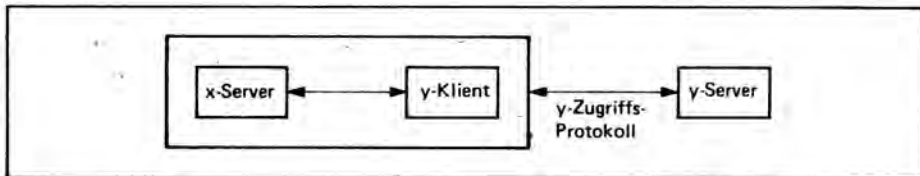
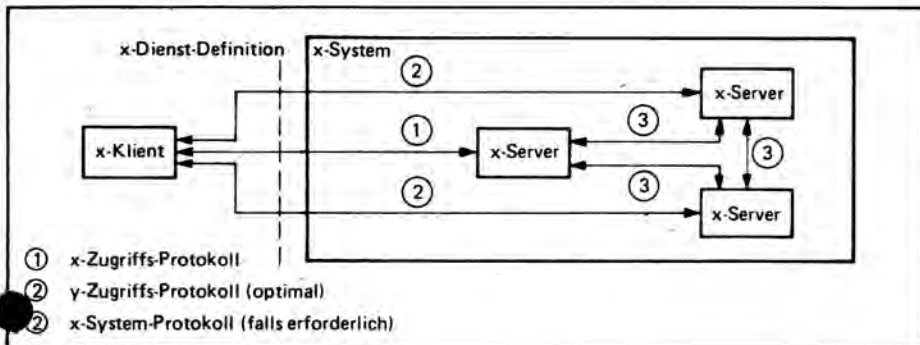


Abb. 2b Multiple verteilte Büroanwendung



Verteilte Verarbeitung – Prinzipien und Modelle



- Ein Server tritt als Nutzer eines anderen Servers auf (Abb. 4); Beispiel ist der Zugriff eines Mitteilungs-Transferagenten zu einem Verzeichnis-Server.

- Direkter Datentransfer zwischen zwei Servern auf der Basis der Übergabe von Bezugsinformationen vom jeweiligen Klienten, wobei beide Klienten sich in einem Anwendungsprozeß befinden. Diese Form des Datentransfers wird als Referenced Data Transfer bezeichnet (Abb. 5).

Beispiel ist der Transfer eines Files von einem Fileserver direkt zu einem Drucker-server.

Interaktionstypen zwischen Anwendungen

Aus dem Dargestellten ergeben sich drei verschiedene Formen der Interaktion zwischen Anwendungen (Abb. 6):

- Um eine produktive x-Anwendung zu benutzen, beschafft sich der Nutzer Informationen über eine (unterstützende) y-Anwendung.

Beispiel ist die bereits erwähnte Beschaffung einer Adreßinformation über den Verzeichnis-Dienst, die beim Senden eines Dokuments über einen Mitteilungsdienst benötigt wird.

- Ein x-Server nutzt einen beigefügten y-Klienten, um über das y-Zugriffs-Protokoll Dienste des y-Servers zur Bereitstellung des x-Dienstes in Anspruch zu nehmen.

Beispiel ist ein Drucker-Server, der sich an den Zeit-Server wendet, um das auszudruckende Dokument mit einem Zeitstempel zu versehen.

- Ein Nutzer instruiert über die x- und y-Klienten die x- und y-Server, einen direkten (referenced) Datentransfer durchzuführen.

Tab. 2 gibt an, welche Interaktionstypen zwischen den einzelnen Anwendungen in Betracht gezogen werden können.

Abb. 3 x-Anwendungssystem mit mehreren x-Servern

Abb. 4 Ein Server als Nutzer eines anderen Servers

Abb. 5 Referenced Data Transfer

Abb. 6 Interaktionstypen

Verteilte Verarbeitung – Prinzipien und Modelle

x-Anwendung	y-Anwendung						
	Verzeichnis	Authentification	Bezugszeit	Mitteilungs-speicher	Ablage und Wiederauffinden	Drucken	Mitteilungs-transfer
Verzeichnis	x	1,2	1,2				
Authentication	1,2	x	1,2				
Bezugszeit			x				
Mitteilungs-speicher	1,2	1,2	1,2	x	3	3	2 (P3)
Ablage und Wiederauffinden	1,2	1,2	1,2	3	2,3	3	
Drucker	1,2	1,2	1,2			x	
Mitteilungs-transfer	1,2	1,2	1,2				(P1)

Weitere Konzepte

Das Modell für verteilte Büroanwendungen enthält weiterhin Namensbetrachtungen, Schutzkonzepte und Richtlinien für den Protokollentwurf.

Referenced Data Transfer (RDT)

An einem „Referenced Data Transfer“ sind drei Parteien beteiligt (Abb. 7):

- ein Initiator, der den Informations-transfer veranlaßt,
- eine Quelle, wo die Informationen gegenwärtig gehalten werden,
- eine Senke, zu der die Informationen übertragen werden sollen.

Der Initiator nutzt ein anwendungsspezifisches Zugriffsprotokoll, um in der Quelle die zu übertragenden Informationen auszuwählen. Bei einem direkten Datentransfer wäre der Initiator gleichzeitig die Senke und die ausgewählten Informationen würden zu ihm übertragen.

Beim RDT spezifiziert der Initiator, daß er von der Quelle anstelle der eigentlichen Informationen eine RDT-Reference erhalten möchte. Diese RDT-Reference, die global eindeutig ist, stellt die Quelle bereit. Der Initiator nutzt dann ein anwendungsspezifisches Zugriffsprotokoll, um der Senke die zu übertragenden Informationen zu übergeben. Beim RDT übergibt der Initiator anstelle der eigentlichen Informationen die zuvor von der Quelle erhaltene RDT-Reference an die Senke. Die Senke nutzt diese RDT-Reference, um eine Transfer-Operation von der Quelle auszulösen.

Der RDT erfordert die Integration zusätzlicher Funktionalität in die Quelle und die Senke:

– eine Quelle muß in der Lage sein, anstelle der eigentlichen Informationen eine RDT-Reference bereitzustellen,

– eine Senke muß in der Lage sein, anstelle der eigentlichen Informationen eine RDT-Reference zu übernehmen,

– eine Senke muß eine Transfer-Operation auslösen können,

– eine Quelle muß eine Transfer-Operation ausführen können.

Die gegenwärtig von der ISO spezifizierten verteilten Büroanwendungen *Mitteilungs-Verwaltungssystem* /3/ (als Quelle bzw. Senke sind die Mitteilungstransfer-Agenten bzw. die Mitteilungsspeicher zu betrachten) und *Verzeichnis-System* /4/ (als Quelle bzw. Senke ist der Directory-System-Agent anzusehen) enthalten diese Funktionalität noch nicht.

Zwischen dem Zeitpunkt der Bereitstellung der RDT-Reference und dem Transfer-Zeitpunkt können andere Nutzer eventuell auf die gleichen Informationen Bezug nehmen und sie ändern oder auch löschen. Es wurden deshalb drei Dienstqualitäten definiert, die eine Quelle bereitstellen kann:

Niveau 1:

Die Informationen können in der Zwischenzeit geändert oder auch gelöscht werden

Niveau 2:

Änderungen werden der Senke angezeigt und es ist der Senke die Entscheidung vorbehalten, die geänderten Informationen zu verarbeiten

Niveau 3:

Wie Niveau 2, jedoch werden in einem fixierten Zeitintervall die Original-Informationen gesichert.

Der Initiator fordert von der Quelle ein

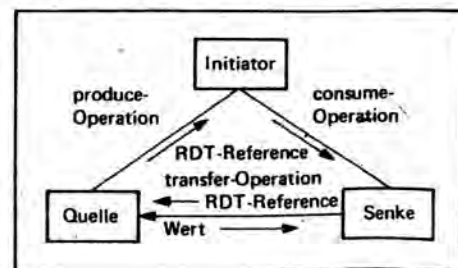
Tab. 2 Beispiel einer Matrix für Anwendungs-Interaktion

bestimmtes Niveau, die Quelle antwortet mit dem Niveau, das sie bieten kann. Der RDT enthält weitere Funktionalität, wie die Möglichkeit der Anforderung einer Veränderung der Dienstqualität durch die Senke bei der Quelle, die Nutzung einer RDT-Reference für mehrere Übertragungen, Authentifikation der Senke bei der Quelle über die Berechtigung der Benutzung einer RDT-Reference.

Abspeicherung und Wiederauffinden von Dokumenten

Im SC18 der ISO gibt es ein Arbeitsthema, das auf die Standardisierung einer Anwendung zur Abspeicherung und zum Wiederauffinden von Dokumenten gerichtet ist, die sich in das Modell für verteilte Büroanwendungen einordnet /5/. Vorarbeiten dazu wurden, wie in vielen anderen Fällen, von der Vereinigung (West-) Europäischer Computerhersteller ECMA geleistet. Ausgangspunkt war die Erkenntnis, daß die von der ISO standardisierte Anwendung für

Abb. 7 Funktionelles Modell für Referenced Data Transfer



Fileübertragung, -zugriff und -manipulation FTAM in der Semantik den Anforderungen der Büroumgebung nicht entspricht und sich nur schwer mit anderen verteilten Büroanwendungen zusammenbringen läßt.

Beim Document Filing and Retrieval (DFR) geht es um die Speicherung und Verwaltung von Bürodokumenten und Gruppen von Dokumenten und deren einfaches Wiederauffinden für eine spätere Nutzung. Entsprechend dem Modell für verteilte Büroanwendungen enthält die DFR-Anwendung einen DFR-Klienten und einen DFR-Server.

Jedes Dokument hat eine Anzahl von Attributen, die zur Auswahl eines Dokuments aus dem im DFR-Server integrierten Dokumentenspeicher verwendet werden können. Über die im Dokumentenspeicher mögliche Gruppenbildung können Dokumente zusammengefaßt werden zu Mappen, Akten usw. Über Referenzen können Dokumente mehreren Gruppen zugeordnet werden, wobei nur in einer Gruppe das eigentliche Dokument gespeichert ist. Es sind drei Dienstqualitäten bezüglich der Referenzen definiert, deren Bedeutung den beschriebenen Niveaus entspricht.

Die wesentlichsten Operationen sind Anlegen eines Dokumenteneintrags, Streichen eines Dokuments, Kopieren eines Dokuments, Verschieben eines Dokuments, Wiederauffinden eines Dokumenteninhalts, Ersetzen eines Dokumenteninhalts. Weitere Operationen beziehen sich auf Gruppen, Referenzen und Attribute.

Die für Dokumente zugelassenen Attribute sollen als Untermenge die entsprechenden Attribute der ebenfalls von der ISO erarbeiteten Büro-Dokumentenarchitektur enthalten.

Drucken von Dokumenten

Die Ausarbeitung einer sich in das Modell für verteilte Büroanwendungen einordnenden Anwendung für das Drucken erfolgt, wie das DFR, im SC18 der ISO und ebenso auf der Basis von Vorarbeiten der ECMA /6/. Die Anschaffungskosten moderner Drucker, die mit großer Geschwindigkeit und hoher Qualität drucken, sind hoch und erfor-

dern die Benutzung durch viele Nutzer. Die Drucker-Anwendung enthält einen Drucker-Klienten und einen Drucker-Server.

Als Operationen stehen Drucken eines Dokuments, Modifizieren eines Druckauftrages, Lesen des Auftragsstatus, Abbrechen eines Auftrages, Auflisten der Aufträge, Lesen des Status des Drucker-Servers, Lesen des Profils des Drucker-Servers, Lesen der Charakteristika des Drucker-Servers sowie administrative Operationen zur Verfügung.

Die Operation zum Drucken eines Auftrages enthält Parameterklassen, mit denen spezifiziert werden:

- Auftragsparameter (Titel, Ersteller u. a.),
- Handlungen nach Abschluß des Auftrags,
- Handlungen, die bei besonderen Ereignissen während des Druckprozesses durchzuführen sind,
- Instruktionen für die Reihenfolgebearbeitung, z. B. Priorität,
- Drucker-Identifikation, um einen Drucker auswählen zu können, wenn der Drucker-Service alternative Drucker bereitstellt,
- Instruktionen zur Dokumentenherstellung,
- Beschreibung des Dokuments einschließlich dessen Charakteristika,
- Instruktionen für eine spezielle Bearbeitung, z. B. Drucken auf ein Formblatt,
- Schutz und Zugriffssteuerung.

Das Zugriffsprotokoll geht nicht von einem bestimmten Druckformat aus, es transportiert verschiedene Druckformate.

Weitere Büroanwendungen

Als weitere sich mit dem Modell für verteilte Büroanwendungen in Einklang befindliche Anwendungen werden die Mitteilungsverwaltungssysteme /3/ und der Verzeichnis-Dienst /4/ angesehen, die bereits weitgehend standardisiert sind. Sie wurden bereits an anderer Stelle ausführlich beschrieben /7,8/.

Schlußbemerkungen

Mit der Schaffung OSI-gerechter Software bis zur Schicht 7 und insbesondere

der Realisierung eines *Remote Operation Service* /9/, mit der Realisierung eines Mitteilungs-Verwaltungssystems und eines Verzeichnis-Dienstes werden auch in der DDR Voraussetzungen für einen Übergang zu standardgerechten verteilten Büroanwendungen geschaffen. Sie sind zukünftig durch weitere, sich noch im Stadium des Entwurfs und der Standardisierung befindliche Büroanwendungen zu ergänzen.

Literatur:

- /1/ Information Processing Systems – Text Communication – Distributed-Office-Applications Model
- Part 1: General Model, DP 10031-1
- Part 2: Referenced Data Transfer, DP 10031-2
- /2/ Information Processing Systems-Open Systems Interconnection – Basic Reference Model, ISO 7498
- s. a.: K. Garbe, V. Heymer, Das OSI-Referenzmodell – Kommunikation zwischen offenen Systemen, *rechen-technik/datenverarbeitung*, 25 (1988) 7, S. 5–9
- /3/ Information Processing – Text Communication-Message Oriented Text Interchange System, ISO DP 10021/1-7
- /4/ Information Processing Systems – Open Systems Interconnection – The Directory ISO DP 9594/1-8
- /5/ Document Filing and Retrieval, OSN: The Open Systems Newsletter, February 1988
- /6/ A new Application for Printing, OSN: The Open Systems Newsletter, Marc/April 1988
- /7/ F. Försterling, I. Müller, C. Sattler, Mitteilungs-Verwaltungssysteme, *rechen-technik/datenverarbeitung*, 25 (1988) 7, S. 22–25
- /8/ I. Kerlikowski, A. Wurzinger, Verzeichnis-Dienst, *rechen-technik/datenverarbeitung*, 26 (1989) 1, 24–26
- /9/ W. Rommel, Die Anwendungsschicht, *rechen-technik/datenverarbeitung*, 25 (1988) 7, S. 17–20

♦ **Hinweis:**
**Ab sofort können auch
in edv-aspekte
Anzeigen veröffentlicht
werden!**

Datenbasis-Entwurfs-System DES

Prof. Dr. Gerd Rossa, Wilhelm-Pieck-Universität Rostock
Institut für sozialistische Wirtschaftsführung, Informatik-Labor

Motivation und Grundphilosophie des Systementwurfs

Entwicklungsziel war ein Datenverwaltungskern mit weitgehenden Eigenschaften eines Datenbankbetriebssystems.

Auf der Basis dieses Kernes soll es möglich sein, sowohl Applikationen mit typischen Datenverwaltungsfunktionen effektiv und kompatibel zu produzieren, als auch ein datenbankintegriertes wissensbasiertes System zu schaffen. Da der Kern einheitlich in allen Applikationen enthalten ist, ist ein integrierter Zugriff zu allen Daten auch von später zu erstellenden Programmen möglich. Dieser Kern muß derart ausbaubar sein, daß er folgenden Anforderungen genügt:

- einsetzbar für Non-Standard-Datenbank-Probleme
- netzwerkfähig
- Schnittstellen zu anderen Standardsystemen sowie zu PASCAL und C
- zweistufige Datenbeschreibung (Data-Dictionary) zur Kopplung zu Expertensystemen.

Gegenstand unserer Forschung zu dem Problem verteilter Datenbanksysteme sind folgende Schwerpunkte:

- Charakteristik verteilter Datenbanksysteme
 - verteilte Transaktionsverwaltung
 - Synchronisation
 - Commit-Verarbeitung
 - Fehlerbehandlung
 - Kommunikation und Netzwerkverwaltung für verteilte Datenbanksysteme
 - die Rolle replizierter Daten in einer verteilten Datenbank
 - Anfrageoptimierung
 - Datenbankentwurf und Administration in verteilten Datenbanksystemen
 - Kopplung von Datenbanksystemen mit Expertensystemen.
- Entsprechend diesen Bedingungen ist das konkrete Datenmodell entworfen worden.

Datenmodell des DES

Die drei Standard-Datenmodelle entsprechen gewissen idealisierten Anwendungsfällen. Jede konkrete Problematik läßt mehr oder weniger die Nachteile dieser Idealisierung deutlich werden.

Alle objektorientierten Funktionen müssen oberhalb des DBS durch den Anwender organisiert werden. Dies ist als Hauptmotiv einer Reihe von Ansätzen zur Erweiterung der Standardmodelle anzusehen.

Objektorientierte Datenbanksysteme gestatten es, mit beliebig komplex aufgebauten Informationseinheiten der interessierenden Anwendungswelt in dedizierter Weise umzugehen. Sie erweitern hierzu die heute üblichen (z. B. relationalen) Datenmodelle und übernehmen Eigenschaften objektorientierter Programmiersprachen wie Smalltalk. In den meisten Fällen wird deshalb versucht, das relativ praktikable Relationenmodell durch hierarchisch organisierte Beziehungen zwischen den Relationen zu erweitern, um eine Objekt-Behandlung auf der Ebene des logischen Modells anzubieten.

Begriffe wie *Wurzel* und *Referenz* deuten auf die Idee der Überlagerung des Relationsmodells durch eine hierarchische Struktur hin.

Das dem logischen Schema zugrunde liegende Datenmodell muß es dem Anwender ermöglichen, die Informationsstrukturen eines konkreten Problems möglichst adäquat abzubilden.

Da die drei klassischen Datenmodelle den vorliegenden Problemklassen nur unvollkommen entsprechen, wurde für das DES ein erweitertes Relationenmodell entworfen.

Wesentlichster und auch umfassendster Teil ist die Strukturbeschreibung, die sich in dem Data Dictionary (DDS) widerspiegelt. Sie ist zweistufig und enthält Elemente der Manipulation, Integrität und Distribution. Die zwei Ebenen teilen sich in syntaktische und semantische Modellebene auf.

Die zweite Ebene des DDS hat insbesondere den Zweck, KI-Komponenten über dem Datenbasiskern zu installieren. Dies vor allem mit dem Ziel, spezifisches Wissen aus konventionell verwalteten und existierenden Datenbeständen zu extrahieren. Das derartig konzipierte Datenmodell ermöglicht den Einsatz des DB-Kernes für Probleme, die folgende Erweiterungen voraussetzen:

- Definition komplexer Datentypen (Objekte, Objektklassen)
- Versionsverwaltung von Objekten
- Unterstützung temporaler Probleme
- Wahlweise Festlegung der Gültigkeit der Integritätsbedingungen (Ubiquitätsprinzip oder Need-to-Know-Prinzip)
- Verwaltung unscharfer Informationen (vages Wissen).

Im DES wird die Objektdefinition durch drei verschiedene Typen von Referenzen gegeben sein:

1. 1 : 1-Referenz
2. 1 : n-Referenz
3. 1 : n-Referenz + Attribute der Referenz.

Diese Referenzen werden separat verwaltet und selbstverständlich in Updates einbezogen.

Mit ihrer Hilfe lassen sich komplexe Objekte definieren und als Ganzes verwalten. Es ist sowohl möglich, Einzelobjekte zu definieren als auch das Objekt als strukturierten Datentyp zu verwenden. Für Objekte existieren spezifische Manipulationsmodule, die dem Benutzer mächtige Kommandos in die Hand geben.

Eine Referenz dritter Ordnung kann z. B. ein Liefervertrag sein, der mit dem Kunden aus einer Relation als Wurzelement eine 1:n-Referenz auf die Relation der (zu verkaufenden) Produkte abbildet und die Attribute dieser Referenzen der jeweilige Produktpreis (für diesen speziellen Kunden) und die vereinbarte Stückzahl sind.

Das Datenbeschreibungssystem (DDS) des Datenbasis-Entwurfs-Systems (DES)

In der ersten Schicht des DDS spiegelt sich das logische Datenmodell wider. Es setzt sich zusammen aus

- Dateibeschreibung
- Feldbeschreibung
- Zusatzangaben zu den Strukturelementen der Datenbasis.

Strukturelemente der Datenbasis sind:

- Datenbasis
- Objekt
- Record
- Feld.

Als Datentypen sind realisiert (*) bzw. vorgesehen:

- Einfache Datentypen
 - * (R) Real
Wertebereich: $10e-39$ bis $10e+38$,
11 signifikante Stellen
 - * (I) Integer
Wertebereich: -32768 bis 32768
 - * (B) Byte
Wertebereich: 0 bis 255
 - (H) BusinessInt
Wertebereich: -2.147.483.648 bis
2.147.483.648
 - (L) Boolean
 - (T) Time
 - * (D) Date
 - * (F) String mit vereinbarter fester Länge
- Erweiterte Datentypen
 - * (V) String variable Länge (max. 255)
 - (A) String fix - automatisch codiert
 - (C) String fix - vorgegebener Code
 - * (W) String variabler Länge mit Wiederholungen
 - * (S) Text - String mit Superlänge (ohne Begrenzung)
 - * (G) Bulk - beliebiger Hexcode (Programme, Graphik usw.)
 - (M) Maske
 - (K) match-code - künstliches Feld aus Kombinationen vorhandener Felder
- Semantische Datentypen (abgeleitet aus obigen Datentypen)
 - * (a) 3-dimensionales Array
 - * (t) Tabelle
 - * (f) Frame
 - * (s) Slot
 - (b) Datenbereichs-Typ
 - (m) Methode.

Die zweite Schicht des DDS verwaltet die semantischen Metadaten, die das Wissen über die *Miniwelt* darstellen, die durch die Datenbasis repräsentiert wird. Diese *Miniwelt* besteht aus strukturierten Objekten und Attributen, die diese und ihre Struktur beschreiben. Objekte lassen sich sowohl gerichtet (hierarchisch) als auch ungerichtet (netzwerkartig) zusammenfassen. Diese Korrespondenzen können durch Wahrscheinlichkeiten bewertet werden. Ziel des semantischen DDS ist es einerseits, eine allgemeine Schnittstelle zu wissensbasierten Systemen zu schaffen und andererseits die Voraussetzungen für (in einem Computernetzwerk) kooperierende Expertensysteme zu schaffen.

Eine dritte integrierte Komponente des DDS beschreibt sowohl EDV-technolo-

gische Sachverhalte als auch durch das System zu sichernde Konsistenz- und Reorganisationsbedingungen.

Zu diesen Kategorien gehören:

- A. - Prüfungsart
 - logische Prüfbereiche und Prüfmasken
 - Maßeinheit
- B. - Gültigkeitsdauer
 - Priorität
 - Archivierungsdauer
 - Perioden der Änderung
 - Perioden der Erfassung
 - Perioden der Ausgabe
 - Perioden der Verarbeitung
 - Perioden der Löschung
 - Quelle
 - betriebliche Strukturebene
 - Ablage der Urbelege
 - Erfassungsart
 - Nutzungsart (Stammdaten, Bewegungsdaten, Prozeßzustandsarten)
- C. - Fixe Feldattribute (drei Optionen, z. B. Behandlung der Nullwerte)
 - Reorganisation (vier Optionen)
 - Behandlung der Löschungen (drei Optionen)
 - Back-Up-Behandlung (vier Optionen)
 - Code of rewrite (Need-to-know-Prinzip in verteilten Systemen)

DES als verteiltes System (VDB)

Viele etablierte Datenbanksysteme sind in ihrem Systemkonzept als Einplatzsysteme entworfen und implementiert worden. Sie wurden dann nachträglich nach dem Serverkonzept netzwerkfähig gemacht (z. B. dBASE, INGRES). Diese Entwicklung ist logisch und in lokalen Netzen praktisch bedeutsam.

Wissenschaftlich von hohem Interesse ist jedoch der umgekehrte Ansatz, der prinzipiell von einem verteilten System ausgeht, und den Einplatzbetrieb als einen Spezialfall betrachtet. Die Integration dieser Verteiltheit-Eigenschaft wird im DES u. a. in folgenden Aspekten sichtbar. Der DB-Designer kann explizit zwischen fünf Arten von Dateien wählen, die durch den LAN-Code einer Datei beschrieben werden:

- Globale Files
- Verteilte Files Typ 1 (identische Struktur)

- Verteilte Files Typ 2 (unterschiedliche Struktur)

- Private Files Typ 1 (dem Netz zugänglich)

- Private Files Typ 2 (dem Netz unbekannt).

Die globalen Files werden nur auf einem Knoten geführt und stehen dem Netz im Sinne eines Fileservers zur Verfügung.

Die verteilten Files vom Typ 1 sind Dateien mit derselben Struktur, die auf verschiedenen Computern verwaltet werden. Der Typ 2 der verteilten Dateien ist dadurch gekennzeichnet, daß sie sachlich korrespondieren. Diese sachlichen Zusammenhänge spiegeln sich in zumindest einem gleichen Datenfeld wieder. Für beide Fälle kann eine Personal-Datei als erklärendes Beispiel dienen. Der Typ 1 ist denkbar im Falle von Betriebsteilen, die ihren Personalbestand separat verwalten und für die die übergeordnete Leitung über das Netz Gesamtanalysen vornehmen kann. Der Typ 2 kann vorliegen, wenn sachlich unterschiedliche Informationen des Personalwesens verwaltet werden. Z. B. können in einer Abteilung die Lohnbearbeitung erfolgen und in einer anderen der Krankenbestand, die Betriebszugehörigkeit oder die Zahl der Kinder der Angestellten von Interesse sein. Diese Abteilungen benötigen differenzierte, aber sachlich zu einer Person gehörende Informationen.

Dieser Problembereich kann auch zur Demonstration der Anwendung der superlangen Felder dienen, wenn man sich vorstellt, daß eine Abteilung die Personalakten verwaltet und im Sinne einer automatischen Ablage alle zum Personal gehörenden maschinenlesbaren Dokumente (wie Beurteilungen oder Lebensläufe) sich als einfache Felder innerhalb einer strukturierten Datei darstellen.

Die privaten Files vom Typ 1 werden auf dem betreffenden Computer geführt und stehen abhängig vom gesetzten Zugriffsschutz dem Netz zur Verfügung. Auf den Typ 2 der privaten Files hat das Netz keinen Zugriff (weder auf Daten noch auf die DDS).

Diese Fileklassifikation erlaubt es ei-

nem verteilten Datenbank-Betriebs-System (VDDBS) in Verbindung mit einigen zusätzlichen Informationen aus dem DDS, eine verteilte Datenbank zu verwalten. Das Entwurfskonzept dieses VDDBS ist in *Prozessmodell eines VDDBS* in diesem Heft ausführlich beschrieben.

Transparenz im Netz

Transparenz für den Nutzer im allgemeinen Sinne bedeutet letztlich nur, daß der Nutzer alle zur Erledigung seiner Aufgaben notwendigen Daten an seiner Arbeitsstation zur Verfügung gestellt bekommt. Ein solches System erfordert eine verteilte Datenbank, die wiederum den verteilten Zugriff auf der Basis von Netzwerk-Kommunikationssoftware voraussetzt. Um es exakter zu definieren: eine verteilte Datenbank erfordert (im Gegensatz zum verteilten Zugriff) nicht die Kenntnis der physischen Adresse des betreffenden Knotens im Netz. Diese Arbeit wird dem Nutzer durch das verteilte Datenbankbetriebssystem (VDDBS) abgenommen. Bei Anforderung an ein Remote-File muß das VDDBS sich die notwendigen Informationen aus dem DDS beschaffen und dem Netzwerk-Kommunikations-System übergeben.

Das *Beschaffen* der Systeminformationen ist eine zweite Komponente verteilter Systeme, denn wenn die Daten verteilt sind, muß das DDS ebenfalls verteilt sein bzw. einem verteilten Zugriff offen stehen.

Hierfür gibt es prinzipiell drei Möglichkeiten:

1. Alle Knoten kennen das DDS der gesamten Datenbasis als identische Duplikate des DDS einer verteilten Datenbank.
2. Eine ausgezeichnete Station enthält das DDS der gesamten verteilten Datenbank. Die lokalen Stationen verwalten nur ihre eigene Sicht auf die Datenbank, kennen aber die Adresse des Masters dieses logischen Sternsystems.
3. Bei erforderlichen Remote-Zugriffen startet der betreffende Knoten eine Anfrage (Polling) ans Netz, um den Knoten zu ermitteln, auf dessen Daten zugegriffen werden muß.

Eine relativ isolierte und zeitlich stabile

Aufgabe empfiehlt eine Organisationsvariante des ersten Typs. Die verteilte Datenbank wird entworfen, das DDS für jeden Knoten festgelegt und daraus die entsprechende Netzwerk-Kommunikation abgeleitet und installiert. Der Vorteil liegt in dem direkten (einstufigen) Zugriff auf jeden beliebigen Knoten im Netz. Der Nachteil wird daraus natürlich auch deutlich: Jede Änderung in der Datenstruktur bedeutet ein Aktualisieren des DDS auf *jedem* Knoten der verteilten Datenbank.

Die zweite Variante erlaubt eine flexible und offene Architektur der verteilten Datenbank, erfordert jedoch mehr Aufwand in Systementwurf und -wartung sowie in der Kopplung zum Netzwerk-Kommunikationssystem, da sich dann das DDS als eine der eigentlichen Datenbanken übergeordnete Datenbank darstellt.

Die dritte Variante erfordert eine spezifische Netz-Kommunikation, da hierfür jeder Netzknoten sowohl als Server als auch als Client arbeiten muß, sowie ein effektives Broadcast-Kommando im Netz.

Das DES als offenes System loser Kopplung

Die Arbeiten am DES sind konzeptionell so angelegt, daß es nach *Wedekind* die Integrationsstufe 4 – eine *Verteilte Datenbank* – ermöglichen soll.

Dies wird durch folgende Komponenten versucht zu erreichen:

- Ein breites Angebot der verschiedensten Datentypen, bis hin zum Objekttyp, ermöglicht weitgehende Anpassung an die Semantik des Problems.
 - Die Berücksichtigung der Verteilbarkeit realer Datenbestände im konzeptuellen Schema ermöglicht das Verwalten einer verteilten Datenbank innerhalb einer heterogenen Netzwerkarchitektur (d. h. unterschiedlicher Netz-Software).
 - Eine erweiterbare Menge von DB-Gateways ermöglicht die Einbeziehung von Datenbeständen, die von anderen Systemen verwaltet werden: z. B. dBASE-Dateien, .PAS-Dateien, .FOR-Dateien, TOPAS-Dateien (Mainframe).
- Abhängig von den genutzten Datenty-

pen wirken diese Gateways uni- oder bidirectional.

- Eine PASCAL-Schnittstelle ermöglicht die Nutzung von DES-Dateien in speziellen Programmmodulen.

- Die Konsistenz und Integritätsbedingungen sind in weiten Grenzen durch den Nutzer definierbar.

Unter loser Systemkopplung wird insbesondere verstanden, daß die Daten an den Knoten gespeichert werden, wo sie am häufigsten benötigt werden. Dieses Grundprinzip verteilter Datenbanken minimiert die verteilten Zugriffe.

Ein weiterer Aspekt loser Kopplung ist die kontrollierte Ausschaltung des Ubiquitäts-Prinzips in Richtung des Need-to-Know-Prinzips.

Über die nutzerdefinierten Parameter des DDS sowie die dynamischen Dateiparameter lassen sich entsprechende Kontroll-Moduln aktivieren.

Das Prinzip der losen Kopplung ermöglicht eine flexible Sicht auf den umgebenden Datenbestand, denn über der direkten Ebene der verteilten Datenbank ist eine Kommunikation zwischen verschiedenen verteilten Datenbanken möglich. Realisiert wird diese Sicht durch einen verteilten Zugriff auf DDS verschiedener verteilter Datenbanken.

Praktisch ist dadurch dem konzeptuellen Schema eine zusätzliche Hierarchieebene – die Datenbasis – aufgesetzt worden. Datenbanken können untereinander Dateien austauschen und integriert bearbeiten.

Das Prinzip der Datenverteilung bei loser Systemkopplung (Need-to-Know) besagt, daß Versionsdaten (bedingtes Copy) auf den Knoten (Remote Station) für den Zeitraum *t* ein isoliertes Eigenleben führen.

Danach werden sie (je nach ihrem *Code of rewrite*)

– in die Datenbasis zurückgeschrieben (Fall A)

– oder der Replikationsdatenbestand wird nur synchronisiert (Fall B).

– Der Fall (C), wenn die Daten nur abgesetzt aber nicht wieder zurückgeschrieben werden, muß im DES nicht vom System unterstützt werden, da es auf der Beschreibungsebene durch Definition des Anwenders im DDS gelöst

wird. Diese Variante entsteht dann, wenn aus der Datenbasis ein Datum übergeben wird, das dann in diesem Knoten eigenständig existiert.

Durch die Verteilung der Datenbank auf die funktionelle Struktur des Prozesses wird die Möglichkeit geschaffen, daß der Designer im Schemaentwurf Aspekte des Need-to-Know-Prinzips festschreiben kann.

Diese Möglichkeit eröffnet dem System u. a. ein weites Anwendungsfeld im rechnergestützten Produktionsbereich.

Konsistenz der verteilten Datenbasis

Lose Kopplung bedeutet für die Konsistenz und Integrität eine Aufteilung in globale und verteilte Mechanismen (zu ihrer Sicherung).

Die Beachtung der Konsistenz von ausgelagerten Datenbeständen wird durch die Knoten in den Zeitintervallen der Systemisolierung selbständig und unabhängig vom verteilten Datenbankbetriebssystem gesichert. Zu den Zeitpunkten der Synchronisation wird dann abhängig vom gesetzten *Code of rewrite* die Konsistenzkontrolle wieder dem Gesamtsystem übergeben.

Diese Spezifik betrifft natürlich nur echte verteilte Dateien (Typ 1 oder 2 bzw. LAN-Code 2 oder 3). Für globale Dateien, die verteiltem Zugriff offenstehen, wird durch Sperrmaßnahmen gesichert, daß zeitgleiche Anfragen dieselbe Sicht auf die Daten haben (konkurrierendes Update; vgl. Prozessmodell).

Die Replikationen lassen sich durch den Dateityp mit dem LAN-Code = 2 (verteilte Files, Typ 1) sehr leicht behandeln, denn es ist nicht notwendig vorgeschrieben, daß die Dateien gleicher Struktur auf verschiedenen Knoten unterschiedliche Ausprägungen haben müssen. Sie können für Recovery-Maßnahmen im Prinzip auch einfache Copien sein.

Das automatische Nachführen von Replikationen ist zur Erhöhung der Recovery-Möglichkeiten bei Systemausfall zwar oft wünschenswert, belastet aber das Netz u. U. ganz wesentlich und verringert damit die Leistungsfähigkeit des Anwendungssystems.

Das Konzept überläßt es also dem Nut-

zer, eine dem Problem adäquate Betriebsart der verteilten Datenbank zu wählen.

Das ausgefeilte DDS bietet alle Möglichkeiten, Module der Integritätskontrolle, Protokollierung, Recovery und Sperrmechanismen einzusetzen.

Die Fragmentierung einer Relation ist sowohl horizontal als auch vertikal möglich und durch zusätzliche Filterung (SELECT) auf die Replikation zurückführbar:

– Horizontale Fragmentierung:

Auswahl einer Teilmenge von Datensätzen nach logischen Bedingungen

– Vertikale Fragmentierung:

Übernahme einer Teilmenge von Feldern einer Relation.

Durch das Prinzip beider Fragmentierungsarten ist eine kombinierte Fragmentierung als eine sehr komfortable Dateimanipulation im Prozeß der Datenverteilung möglich.

Die Datenmanipulationssprache ANNA des DES bietet diese Möglichkeiten.

Für die Steuerung der Datenverteilung und Konsistenz-Behandlung stehen im DDS u. a. folgende Informationen zur Verfügung:

• Im File der Dateibeschreibung:

– lock/unlock für LAN-Zugriff

– Datum des letzten Update

– Datum des letzten Lesens

– Zahl der Updates seit der letzten Reorganisation

(z. B. Ubiquitätsschwelle)

• Im File der Feldbeschreibung:

– Code of Rewrite

– Priorität

– Gültigkeitsdauer

– Nutzungsart

• Im Datenfile:

– Status (mit Angaben über den letzten aktiven Prozeß).

Zugriffsschutz

Der Zugriffsschutz wird analog UNIX organisiert. In einem für alle verteilten Datenbanken nutzbaren User-File werden folgende Daten verwaltet:

– login-Name

– Passwort

– Gruppennummer

– Name der Datenbasis (für die der Nutzer zugelassen ist).

Im jeweiligen DDS einer verteilten Datenbasis sind für den Zugriffsschutz auswertbar:

• Im File der Dateibeschreibung:

– LAN-Code

– Nummer der LAN-Station

– Zugriffscode

– Identifikation des Eigners der Datei

– Identifikation des letzten schreibenden Nutzers

– Identifikation des letzten lesenden Nutzers

• Im File der Feldbeschreibung:

– Zugriffscode (auf das Feld bezogen)

• Der Eigentümer einer Datei bestimmt bei deren Beschreibung durch das Setzen der Schutzbits, die die Nutzungsart und Nutzergruppe steuern, den wesentlichen Teil des Zugriffsschutzes.

Es ist klar, daß die Organisation eines Zugriffsschutzes in dieser Weise gewisse praktische Realisierungsprobleme mit sich bringt.

Hardwaremaßnahmen (z. B. ausgebautes oder mechanisch gesperrtes Disk-Laufwerk) und spezielle Betriebssystemvarianten ermöglichen es jedoch auch, in bestimmtem Maße obiges System zu realisieren.

Außerdem erfordert das Entwicklungsziel, eine transparente Datensicht in dem inhomogenen System UNIX/MS-DOS zu realisieren, so daß die UNIX-Zugriffsregelung auch in MS-DOS-Anwenderprogrammen interpretiert werden kann.

Datenbankadministrator (DBA)

Für den DBA existiert eine Reihe von Programmen, die es ermöglichen, Verwaltungsfunktionen wahrzunehmen. Ein Superwort ermöglicht ihm den Zugriff zu diesen Programmen. Dazu gehören:

– Datensicherung (BACK-UP, Replikation)

– Sicherung der Änderungen (UPDATE)

– Entwerfen neuer Dateien bzw. Strukturänderungen (DDS)

Dienstorganisation in lokalen Netzen mittels Fernaufruf

Uwe Schulze, Wilhelm-Pieck-Universität Rostock
Institut für sozialistische Wirtschaftsführung, Informatik-Labor

– Vergabe und Änderung von Zugriffsrechten.

Diese Verwaltungsarbeiten sind im Prinzip bei laufendem Betrieb möglich. Sie können auch durch entsprechende Angaben während des Entwurfes der Datenstruktur durch den Nutzer festgeschrieben werden. Auf diese Weise können der DBA entlastet und die Standardmaßnahmen dem System überlassen werden.

• Back-Up-Standards:

– keine Sicherung im DES-Niveau vorgesehen

– Anlegen einer Back-Up-Datei nach Schließen der Datei bei schreibendem Zugriff

– Anlegen einer Back-Up-Datei nach 10 Prozent Änderungen

– Anlegen einer Back-Up-Datei nach 10 Prozent Änderungen sowie laufende Sicherung aller Änderungen

• Reorganisations-Standards:

– keine Vorgaben

– Physisches Löschen aller logischen Löschungen

– Physisches Löschen aller logischen Löschungen und Sicherung aller logischen Löschungen in einer Archiv-Datei.

Realisierungsstufen

Das DES wird in folgenden Arbeitsstufen implementiert:

1. DES-Kern mit den einfachen Datentypen und einer menügesteuerten Datenmanipulation

2. Einbeziehung ausgewählter erweiterter Datentypen (V.S.G)

3. Implementation der Anfragesprache ANNA

4. Kommandoschnittstelle des DES-Kerns

5. experimentelle Realisierung des Prozessmodells des VDB

6. Implementation zusätzlicher Voreinstellungen der Datenmanipulation.

Prinzip des entfernten Dienstaufrufs

Aus der internationalen Vielfalt an Netzhardware ergibt sich neben einem großen Angebot an netzfähiger Standardsoftware auch die Möglichkeit, eigene Applikationen zur Nutzung in einem lokalen Netz zu schreiben. Da beim Programmierer verteilter Anwendungen i. allg. weder die Erfahrung der Programmierung in verteilter Umgebung, noch die Fähigkeit der Implementation komplizierter Algorithmen zur Netzverwaltung vorausgesetzt werden können, ergibt sich die Forderung nach einer von der konkreten Netzarbeit weitgehend unabhängigen Schnittstelle. Besonders von den Problemen der Synchronisation, der Adress- und Namenverwaltung sowie der Time-Out-Behandlung soll der Anwendungsprogrammierer weitgehend befreit werden. Dabei gestaltet sich gerade die letzte Forderung schwierig, da aufgrund der differierenden Ausführzeiten für Netzdienste nicht festgestellt werden kann, ob noch auf die Ausführung einer Funktion gewartet wird oder etwa ein Netzfehler (Absturz einer Arbeitsstation) vorliegt. Ziel ist es, den netzspezifischen Teil aus Anwenderprogrammen mehr und mehr in die Module zur Unterstützung entfernter Dienste zu verlagern. Die Abhängigkeit der einzelnen Arbeitstationen voneinander soll gering sein und Fehler in anderen Netzknotten den eigenen Programmverlauf möglichst wenig beeinflussen.

Hier sollen die Ziele und Probleme der Gestaltung von komfortablen Zugangsschnittstellen zu entfernten Netzdiensten diskutiert werden.

Die Kommunikation in verteilten Systemen läßt sich in nachrichtenorientierte (message based) und funktionsorientierte (procedure based) Kommunikation unterteilen. Bei ersterer erfolgt ein Verschicken von Nachrichten in beide Richtungen gleichberechtigt über eine SEND- und eine RECEIVE-Funktion. Je nach Spezifikation werden synchrone und asynchrone Kommunikation unterschieden. Der funktionsorientierten Kommunikation liegt die Organisation nach dem Server-Client-Modell (Abb. 1) zugrunde, wie sie vielen verteil-

ten Anwendungen entspricht. Dabei werden im Netz Dienstbringer (Server) und Dienstanwender (Clients) installiert. Diese Dienste können auf Systemebene liegen und File- oder Print-Funktionen ähnlich den lokalen anbieten, aber auch auf Applikationsebene und beispielsweise spezielle Datenbankdienste wie EDIT, APPEND, DELETE liefern. In der Praxis werden i. allg. Server für eine Gruppe von Diensten installiert (Fileserver, Printserver, Datenbankserver etc.).

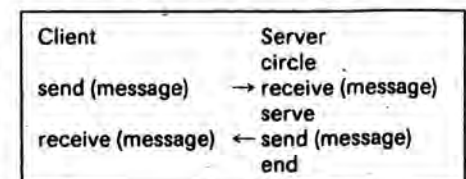
Dem entfernten Dienstaufruf (Remote Procedure Call – RPC) liegt die Idee zugrunde, solche Dienste wie lokale nach einem Unterprogrammkonzept zu nutzen, wie es in PASCAL oder C realisiert wird. Damit ist der Fernaufruf ein Mechanismus zur Realisierung einer hochgelegenen komfortablen und netzunabhängigen Schnittstelle. In einigen Programmiersprachen gehören Konstrukte zur Kommunikation und Synchronisation voneinander unabhängig lauffähiger Programm-Module auch direkt zum Sprachumfang (z. B. Ada).

Die inzwischen allgemein anerkannte Definition findet sich bei /3/: *Remote Procedure Call ist der synchrone Transfer von Daten und Steuerinformationen zwischen Teilen eines über mehrere disjunkte Adressräume verteilten Programms.*

Die Implementation von RPC-Mechanismen ist auf recht unterschiedlichem Niveau möglich. Beispiele dafür sind Betriebssystemdienste (Network File System von Sun) oder Spracherweiterungen (i. allg. Spezialsprachen). Das Binden einzelner Module ist entweder zur Entwicklungszeit oder zur Laufzeit möglich.

Die Anforderungen an entfernte Dienstaufrufe lassen sich wie folgt zusammenfassen:

Abb. 1 Server-Client-Beziehung /1/



Verteilte Verarbeitung – Prinzipien und Modelle

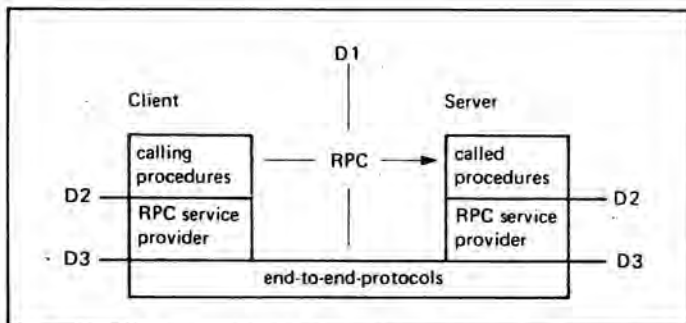


Abb. 2 Einordnung der RPC-Dienste in das OSI-Referenzmodell

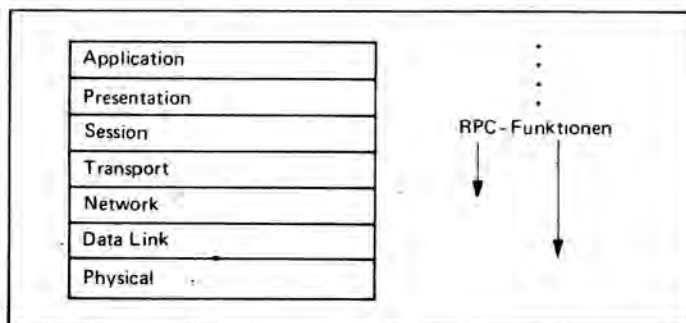


Abb. 3 Schnittstellen im RPC-Modell /4/

- Verbindungen bestehen auf Programmebene
- Konstrukte sind unabhängig von Programmiersprachen
- Dienste und Netzgegebenheiten sind weitgehend entkoppelt
- Anforderungen an Nutzerprogramme sind gering.

Netzverwaltung

Beim Fernaufruf werden im wesentlichen drei Aktionen ausgeführt:

1. Versenden eines Datagramms zur Aktivierung des entfernten Dienstes und Übergabe der Parameter
2. Entfernte Ausführung des Dienstes
3. Empfangen eines Datagramms zur Rückgabe der Programmsteuerung und Übergabe von Parametern.

In /5/ kann eine Anforderung oder eine Antwort zur Übermittlung einer großen Datenmenge aus mehreren Paketen bestehen. Da aber stets das gleiche Wechselspiel erfolgt, ist die Handhabung für den Programmierer einfach.

Prinzipiell lassen sich entfernte Dienstaufrufe auf verschiedenen Ebenen nutzen. Möglich sind ein komfortabler Transportdienst (durch Verkoppeln je einer Sende- und Empfangsfunktion), ein Fortschreiben von Aktionen in einem Sitzungsdienst oder ein zusätzliches gemeinsames Typverständnis auf Darstellungsebene.

In den Normungen der ISO sind RPC-Dienste in Schicht 7 des Referenzmodells eingeordnet und auch die ECMA schlägt Dienste vor /4/, die Elemente der Darstellungsschicht enthalten. Die meisten Implementationen entsprechen diesen Forderungen noch nicht, die Entwicklung wird aber in diese Richtung gehen.

Als Transportdienste können entweder verbindungsorientierte Dienste der Schicht 4 (Transport) des OSI-Referenzmodells oder verbindungslose der Schicht 2 (Data Link) genutzt werden. Die Nutzung verbindungsorientierter

Dienste sichert die Zuverlässigkeit der Übertragung und garantiert, daß Nachrichten nicht verlorengehen, dupliziert oder in der Reihenfolge vertauscht werden. In lokalen Netzen gilt die Linkschicht zur Realisierung des Fernaufrufs als ausreichend zuverlässig. Ein einfacher Sicherheitsmechanismus ist mit dem strengen Wechsel von Anforderungen (request) und Antworten (response) gegeben. Für den Auftraggeber ist das Ergebnis gleichzeitig Quittung für die Ausführung; der nächste Auftrag ist für den Server Quittung der vorangegangenen Antwort. Das Verlorengehen eines Paketes kann nur über einen Time-Out-Mechanismus erkannt werden. In einem solchen Falle muß dabei eine doppelte Ausführung der Funktionen verhindert werden (z. B. Folgenummern für jeden Auftrag).

Durch die Nutzung des Fernaufrufs läßt sich die Entfertheit der Dienste für den Nutzer verdecken; sie können wie lokale Unterprogramme genutzt werden. Allerdings müssen zwei Probleme weiterhin auf Anwendungsebene verwaltet werden:

- Sicherung der Konsistenz entfernter Datenbestände durch das Prinzip atomarer Transaktionen (atomic actions) und zusätzlicher Sicherungsmechanismen
- Behandlung von Netz-Anomalien, die nicht auf Transportebene geklärt werden können (Ausfall eines Netzknotens).

Während bei einer lokalen Prozedur garantiert ist, daß sie genau einmal ausgeführt wird (exactly-once-Semantik) – im Falle eines Absturzes wird das Programm von Anfang an oder an Wiedertrittspunkten neu gestartet – ist dies bei entfernten Prozeduren nicht gesichert. Zuverlässige Transportdienste verhindern zwar das Verlorengehen, Verfälschen und Duplizieren von Nachrichten, im Fehlerfall ist aber nicht feststellbar, ob ein entfernter Dienst voll-

ständig ausgeführt wurde oder nicht. So ist es möglich, daß ein verteiltes Programm z. B. bei Sperrung von Dateien oder Datensätzen mit sich selbst (d. h. mit vorangegangenen Aufrufen) in Konkurrenz tritt. Dieses Beispiel zeigt, daß solche Fehler nur auf einem höheren Niveau (z. B. durch Datenbankaktionen) behandelt werden können.

Hinsichtlich der Synchronisation zwischen rufendem und gerufenem Prozeß lassen sich zwei Arten unterscheiden: Synchroner expliziter und asynchroner impliziter Start /2/.

Beim expliziten Aufruf muß der Dienstbringer (Server) bereits aktiv sein, d. h. auf eine Anforderung aktiv oder passiv (polling) warten. Dazu nutzt er eine RECEIVE- oder ACCEPT-Anweisung, die entweder gepuffert ist oder zyklisch aufgerufen wird und mit der SEND- oder REQUEST-Anweisung des rufenden Prozesses korrespondiert. Keine Kontrolle über die Ausführung des gerufenen Dienstes hat der Server dagegen beim impliziten Start. Die Aktivierung erfolgt wie bei einer Interrupt-Service-Routine, die dann auch notwendige Synchronisationsmechanismen enthalten muß.

Für die Schnittstellengestaltung wird in /4/ das in Abb. 2, 3 gezeigte Modell vorgeschlagen. Dabei ist D1 die Spezifikation der Prozeßkommunikation zwischen rufendem und gerufenem Prozeß. D2 stellt die Schnittstelle zum lokalen RPC-Service und D3 die zum genutzten Transportdienst dar. Damit ist beispielsweise die Nutzbarmachung anderer Netzhardware und Transportdienste möglich.

Bei der Realisierung von RPC-Mechanismen nimmt die Frage nach der Adressierung bzw. Namenverwaltung breiten Raum ein. Grundsätzlich kann man explizite, globale und funktionale Adressierung unterscheiden /2/. In der Praxis finden sich häufig Mischformen oder aufeinander aufbauende Adressie-

rungsarten. Bei der expliziten Adressierung muß der Empfänger ausdrücklich angegeben werden. Häufig muß der Empfänger auch den Absender spezifizieren. Die Kommunikation erfolgt von Prozeß zu Prozeß und ist i. allg. nicht an die Netzknoten gebunden. Die globale Adressierung spezifiziert nicht Prozesse, sondern die Netzknoten bzw. darauf zur Verfügung gestellte Empfangspuffer. Die Verbindung zum Prozeß wird über Tabellen hergestellt. Die komplexeste Art der Adressierung ist die funktionale Adressierung. Dabei wird der gewünschte Dienst identifiziert, unabhängig davon, auf welchem Knoten oder innerhalb welchen Prozesses er zur Verfügung gestellt wird.

Dienste und Prozesse müssen nicht notwendigerweise durch Nummern identifiziert werden, die Übergabe des Namens als Zeichenkette ist oft komfortabler.

Bei Realisierung ortstransparenter Dienste muß über die Namenverwaltung im Netz entschieden werden. Möglich ist ein zentraler Namensdienst (Name-Server), oder ein dezentraler (Ermittlung von Diensten oder Prozessen per Rundruf). Für eine zentrale Organisation spricht ein geringerer Kommunikationsaufwand, für eine dezentrale der Wegfall des Prozesses Name-Server und der Abhängigkeit aller Netzoperationen von diesem Knoten.

Auch hier liefern Mischformen die besten Ergebnisse, z. B. dynamisch organisierte Tabellen, die die Zuordnung Dienst – Netzknoten nur der bereits benutzten Dienste enthält; neue Dienste werden per Rundruf ermittelt.

Für welche Form man sich entscheidet, hängt entscheidend davon ab, ob die angebotenen Dienste und ihre Zuordnung zu Netzknoten vielen Änderungen unterworfen sind oder nicht.

Die Parameterübergabe an entfernte Dienste sollte ähnlich der an lokale Prozeduren gestaltet werden. Die unterschiedlichen Adressräume lassen aber eine Übergabe von Adressparametern (call by reference) nicht zu. Es muß also eine Übergabe als Wert (call by value) in beide Richtungen (call by value-result) genutzt werden. Strukturierte Da-

tentypen (string, array, record) werden i. allg. auch als Adressen vermittelt. Deshalb müssen sie in eine verschickbare Form aufgelöst werden. Dazu dient ein byteorganisierter Pufferbereich. Besonders kompliziert gestaltet sich die Übergabe von Listen und Prozedurparametern.

Für die meisten Anwendungen wird es aber ausreichen, die elementaren Datentypen und ein Byte-Array zur Verfügung zu stellen.

Es sei noch darauf verwiesen, daß die Arbeit mit globalen Variablen nicht möglich ist. Zwischen verschiedenen Darstellungen von Datentypen muß gegebenenfalls eine Konvertierung vorgenommen werden.

Durch die lose Kopplung der Prozesse kommt der Fehlerbehandlung besondere Bedeutung zu. Die auftretenden Fehler müssen jeweils auf Ebene ihrer Entstehung beseitigt werden (Übertragungs-, Netzverwaltungs-, Applikationsfehler). Eine Klassifikation – in 4/ werden vier Klassen vorgeschlagen (Normal, Warning, Abnormal, Error) – erleichtert die Behandlung.

Zusammenfassung

RPC-Dienste lassen sich vielseitig anwenden. Sowohl zur Nutzung vorhandener Dienste, als auch zur festen Verteilung eigener Anwendungen sind sie geeignet. Realisierte Dienste können auf Systemebene liegen oder auch reine anwendungsorientierte Dienste (z. B. auf Datenbankebene) sein. Vorbild für Syntax, Parametervermittlung und Ausführung sollen die jeweiligen lokalen Dienste sein. Eine ortstransparente Organisation der Dienste vergrößert den Implementationsaufwand, aber auch den Komfort.

Zusammenfassend läßt sich der Fernaufruf wie folgt charakterisieren:

- hardwareunabhängiger Netzzugang
- asymmetrisches Protokoll nach dem Server-Client-Prinzip
- request/response Prinzip (je ein Anforderungs- und Antwortpaket)
- kein konkurrender Betrieb von Server und Client (sequentielle Programmfolge)

– synchrone Kommunikation (blockierender Aufruf).

Obwohl die meisten RPC-Musterimplementationen mit UNIX in Verbindung stehen, um die Unterstützung der verteilten Verarbeitung durch das Betriebssystem nutzen zu können, kann der entfernte Prozeduraufruf als Modell zur losen Kopplung voneinander unabhängiger Arbeitsstationen auch als Konzept zur Verbindung klassischer Einplatzstationen (PC-Technik) verwendet werden. Sofern vorhanden, sollten genormte Netzzugänge (z. B. NETBIOS) verwendet werden.

Zur Lösung anspruchsvoller Aufgaben reicht eventuell der Mechanismus des entfernten Dienstaufrufs nicht aus. Ein noch komplexeres Modell für verteilte Verarbeitung wird als Open Distributed Processing bezeichnet. Hier werden folgende Problemklassen diskutiert:

- konkurrente Prozeduraufrufe
- Rufe ohne Ausführbestätigung (Immediate Return)
- Rufe per Interrupt
- weitere Aufteilung einer Anforderung durch den Server (Multi-endpoint-Interactions)
- optimierte Protokolle für Spezialanwendungen.

Literatur

- /1/ S. K. Shrivastava, F. Panzieri: The Design of a Reliable Remote Procedure Call Mechanism. IEEE, Vol. C-31, July 1982
- /2/ Matthias Kaiserswerth: Der Fernaufruf als Betriebssystemdienst. Dissertation, Friedrich-Alexander-Universität Erlangen Nürnberg 1987
- /3/ B. J. Nelson: Remote Procedure Call Carnegie Mellon University Computer Science Report: CMU-CS-81-119, 1981
- /4/ Standard ECMA-127: Basic Remote Procedure Call using OSI Remote Operations. European Computer Manufacturers Association 114 Rue de Rhone – 1204 Geneva (Switzerland)
- /5/ Brent B. Welch: The Sprite Remote Procedure Call System. July 2, 1986 Computer Science Division Electrical Engineering and Computer Science University of California Berkeley, CA 94720
- /6/ J. M. Bacon, K. G. Hamilton: Distributed Computing with RPC: the Cambridge Approach University of Cambridge

Kopplung von DBS und XPS

Semantische Datenbeschreibung

Prof. Dr. Gerd Rossa, Wilhelm-Pieck-Universität Rostock
 Institut für sozialistische Wirtschaftsführung, Informatik-Labor

Leitungsinformationssysteme (LIS) basieren immer auf effektiven Datenverwaltungssystemen mit komfortablen Anfrage- und Recherchemöglichkeiten. Was liegt näher als zu versuchen, diese Möglichkeiten mit der dazugehörigen Datenbasis auch für Expertensysteme (XPS) nutzbar zu machen. Ziel kann es dabei sein, die Regeln und Fakten des XPS mit Hilfe des DBS effektiv zu verwalten oder was noch interessanter ist, neue Aussagen für die Wissensbasis aus der Datenbasis abzuleiten. Aus der internationalen Literatur sind dazu zwei prinzipiell unterschiedliche Wege zu erkennen:

1. Transformation von Klauseln des XPS in Formulierungen von Anfragen an die Datenbasis. Hierfür wird in der Regel die gut standardisierte Anfragesprache SOL als Mittler genutzt.

2. Die Datenbasis wird genutzt, die Datenelemente des XPS zu verwalten. Im allgemeinen reichen dazu die klassischen Typen von DB-Schemata nicht aus. Z. B. kann es darum gehen, sogenanntes *Vages Wissen* zu formulieren und zu speichern.

Ein weiterer Weg scheint ebenfalls erfolgsversprechend zu sein. Wenn ein DBS (Datenbasis-Betriebssystem) einen ausführlichen Datenbeschreibungsteil enthält, kann über diesen eine semantische Datenbeschreibung gesetzt werden, die dann dem XPS erlaubt, qualitatives Wissen aus der DB zu ziehen (vgl. Systembeschreibung DES).

Dem Konzept DES liegt der z. Z. noch spekulative Gedanke zugrunde, auf der Basis verteilter Datenbasen kooperierende XPS auf Mehrrechnersystemen entwickeln zu können.

Die Semantischen Meta-Daten (SMD) beschreiben das Wissen über die *Miniwelt*, die durch die Datenbasis repräsentiert wird.

Die *Miniwelt* besteht aus Objekten, die identisch mit dem strukturierten Objekt-Datentyp des DES sein können. Die Strukturierung kann aber auch erst auf der Ebene der SMD durch Festlegung in einem SLOT erfolgen. Neben der direkten Objektstrukturierung lassen sich weitere (bewertete) Relationen zwischen den Objekten festlegen wie z. B. Zuliefer-

abhängigkeiten, deren Bewertung das Maß der Abhängigkeit des Finalisten von dem konkreten Zulieferer angibt. Die Objekte werden durch Merkmale (Attribute) beschrieben.

Die Merkmale unterteilen sich in:

– *Ordinalmerkmale*, die einer Klassifizierung oder Gruppierung entsprechen (z. B. Kombinat, Stammbetrieb, Betrieb)
 – *Basismerkmale*, die Beschreibungsgrößen darstellen wie Zahl der AK, IWP usw.

– *Abgeleitete Merkmale* ergeben sich aus einer Berechnungsvorschrift (Regel) und dienen einer besseren Beschreibung der Objekte (z. B. Kennzahlen wie AP). Alle Merkmale können in folgenden Intervallen in ihrer Gültigkeit eingeschränkt werden: immer, fast immer, häufig, indifferent, selten, sehr selten, niemals.

Die Merkmale können außerdem tendenziell bewertet werden, d. h. vereinbarungsgemäß ist ein hoher Wert als gut und ein niedriger als schlecht zu bewerten.

Die Bewertungsstufen sind: sehr gut, gut, überwiegend gut, indifferent, sehr schlecht, schlecht, überwiegend schlecht.

Datenstruktur der SMD

Die SMD-Datenstruktur ist in drei Hierarchie-Ebenen gegliedert:

Frame – Slot – Facette

Zu jedem Datenfeld der Datenbasis existiert genau ein *Frame*, der eine gewisse Basisklassifikation des Datenfeldes vornimmt.

Die Feldart bestimmt die semantische Hauptgruppe des Feldes.

Semantische Klassifikation der Daten einer Datenbasis:

0. World als spezifisches Identifikations-Objekt (Domain)
1. Identifikation von Objekten
2. Attribute von Objekten
3. Identifikation von Referenzen zwischen Objekten
4. Attribute von Referenzen
5. Ereignisse
6. abstrakte Daten (Ort, Zeit, Datum)
7. Zuordnungen, Tabellen
8. Erklärungen (Texte, Dokumente)
9. Methoden.

Slots sind die Attribute der Frames, können aber strukturiert sein. Dies steht im Gegensatz zu Relationen in DBS, wo nur atomare Ausprägungen möglich sind.

Strukturierung der Slots sind die *Facetten*, die zusätzliche Angaben zu den Attributen enthalten können. Im trivialen Fall enthalten sie die atomare Ausprägung des Attributes.

Möglichkeiten der semantischen Beschreibung der Daten durch Facetten:

- A,w – Atomare Ausprägung des Attributes
- D – Default Wert
- T – Bedingung für Aktion (Trigger for Message)
- V – Vererbung
- P – prozedurale Angaben (Funktion, Procedure)
- R,w – Regel
- F,w – Fact
- S,w – Strukturinformation (Strukturart, Bezug (Objekt, DES-Feldname)).

Slots sind durch ein Gültigkeitsattribut w ergänzbar.

Tab. 1 Bewertungen (WAHRSCH)

semantische Bewertung		Abhängigkeit/Gültigkeit	
1 = sehr schlecht	0 – 14,3	keine	/niemals 0
2 = schlecht	– 28,6	sehr gering	/sehr selten – 20
3 = überwiegend schlecht	– 42,9	gering	/selten – 40
4 = indifferent	– 57,1	mittlere	/indifferent – 60
5 = überwiegend schlecht	– 71,4	hohe	/häufig – 80
6 = gut	– 85,7	sehr hohe	/fast immer – 99
7 = sehr gut	– 100	absolut	/immer 100

Netzbetriebssysteme für den Verbund von Arbeitsplatzrechnern

Axel Wüstemann, Wilhelm-Pieck-Universität Rostock
Institut für sozialistische Wirtschaftsführung, Informatik-Labor

Dateistruktur – FRAME

STATUS : Byte
DATEINR : Byte
FELDNR : Integer
FELDNAME : Str (16)
NAMESLANG : Str (60)
FELDTYP : Char
FELDLÄNGE : Byte
FELDART : Char

Dateistruktur – SLOT

STATUS : Byte
FELDNR : Integer
SLOTART : Char (A, D, T, V, P, R, F, S,)
WAHRSCH : Byte
SLOT : VarStr
TYP1 : Byte (ordinal, metrisch, gemischt, alphanumerisch)
TYP2 : Byte (Basis-, abgeleitetes Datum)
WERTUNG : Byte (positiv, negativ, normalverteilt, ohne Wertung)
VERWENDUNG : Byte (Fakt, Logistik, Erläuterung)

Dateistruktur – TABELLEN

1 – Tabellennummer Integer
2 – Zeilennummer Integer (Nr. = 0 für Tabellenname)
3 – Inhalt Str (80)

Dateistruktur – Methode

siehe Methodenbanksystem M-BASE (S. 61 in diesem Heft).

Bewertungen (WAHRSCH) w siehe Tab. 1.

Netzbetriebssysteme sind neben den verteilten Anwendungen ein wesentliches Konzept zur Verwaltung von Ressourcen in Arbeitsplatzrechner-Netzen. Mit ihrer Hilfe sind sowohl Nutzer als auch Anwendungsprogramme in der Lage, transparent und quasiparallel auf im Netz verwaltete Ressourcen, insbesondere auf Verzeichnisse, Dateien und Geräte zuzugreifen. Sie haben sowohl für Personalcomputer als auch für Rechner der Workstation-Klasse international eine breite Marktakzeptanz gefunden.

In diesem Artikel soll das Netzbetriebssystem-Konzept mit seinen Vor- und Nachteilen diskutiert und anderen Konzepten zur Gestaltung von Arbeitsplatzrechner-Netzen gegenübergestellt werden.

Verteilte Systeme

Verteilte Systeme bzw. genauer *verteilte Rechensysteme* sind Rechensysteme, die als Ganzes oder teilweise in Teilsysteme aufgeteilt sind. Diese Teilsysteme sind räumlich verteilt und durch ein Kommunikationsnetzwerk, wie z. B. ein LAN, lose miteinander gekoppelt.

Die verteilten Hardware-Komponenten eines verteilten Systems sind die *Netzwerk-Knoten*. Die auf jeweils einem Knoten residierenden Software-Elemente eines verteilten Systems werden als *Instanzen* bezeichnet.

Es ist günstig, von den konkreten Eigenschaften des die Knoten bzw. Instanzen verbindenden Kommunikationsnetzwerkes zu abstrahieren. Das führt auf den Begriff des Kommunikations-Teilsystems.

Unter dem *Kommunikations-Teilsystem* soll ein beliebiges lokales Netz mit dazugehöriger Protokoll-Software verstanden werden. Aus dieser abstrakten Sicht realisiert das Kommunikations-Teilsystem Kommunikations-Ressourcen in Form von Sende- und Empfangskanälen mit bestimmten qualitativen Eigenschaften, wie z. B. einem bestimmten Durchsatz, einer bestimmten Fehlerrate usw. Die Kommunikations-Ressourcen können an einer durch das Kommunikations-Teilsystem erzeugten Netzzugangsschnittstelle von Rechenprozessen

verschiedener Instanzen im Netz angefordert werden, um Daten zwischen ihnen auszutauschen.

Dem entspricht beispielsweise ein LAN, das an einer entsprechenden Schnittstelle die Funktionalität etwa der OSI-Transportschicht (Abb. 1) oder einer höheren OSI-Schicht bereitstellt. Auch das NetBIOS /6/ erzeugt eine Netzzugangsschnittstelle.

Diese abstrakte Sichtweise ist durchaus auch von praktischem Wert: Die meisten Netzbetriebssysteme sind intern so ausgelegt, daß sie unabhängig vom konkreten LAN sind. Damit werden sie zum einen besser portierbar und zum anderen unabhängig von sich schnell ändernden technologischen Gegebenheiten der LAN-Hardware.

Abb. 2 zeigt den Übergang von einem herkömmlichen Rechensystem zu einem verteilten System.

Auf der Basis eines solchen Kommunikations-Teilsystems ergeben sich verschiedene Möglichkeiten zur Verteilung von Instanzen auf die Knoten eines Netzwerkes. Wird ein Anwendungssystem in Teilsysteme (Instanzen) zerlegt, die auf die Knoten eines Netzwerkes verteilt werden, so sprechen wir von einer *verteilten Anwendung*. Es können aber auch Komponenten eines Betriebssystems verteilt werden. Dabei können die beiden Konzepte *Netzbetriebssysteme* und *verteilte Betriebssysteme* unterschieden werden.

Verteilte Anwendungen

Unter einer *verteilten Anwendung* wird ein Anwendungssystem verstanden, daß in Teilsysteme (Instanzen) aufgeteilt ist. Diese Teilsysteme sind auf die Knoten eines Kommunikationsnetzwerkes räumlich verteilt.

Die Instanzen einer verteilten Anwendung kooperieren direkt über die durch das Kommunikations-Teilsystem bereitgestellte Netzzugangsschnittstelle miteinander.

Geht man davon aus, daß für ein Rechnerverbundsystem das Netz in bezug auf die Gesamtleistungsfähigkeit i. a. einen Engpaß bildet, so sollte man die auf diesem Verbundsystem laufenden verteilten Anwendungsprogramme so im-

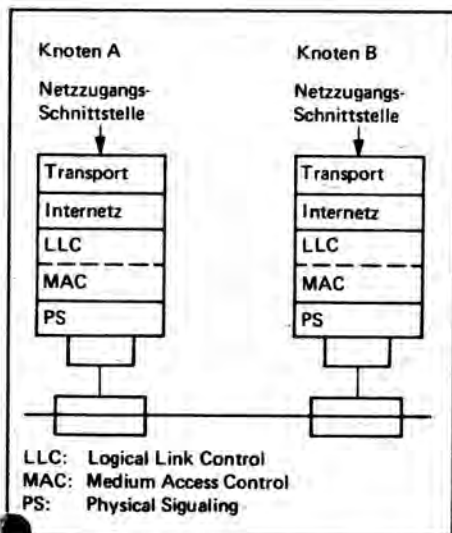


Abb. 1 Beispiel für ein Kommunikations-Teilsystem

plementieren, daß möglichst viel lokale Verarbeitung innerhalb der einzelnen Netzknoten und möglichst wenig Kommunikation zwischen den einzelnen, verteilten Anteilen dieser Anwendung stattfindet.

Mit dem Aufsetzen einer verteilten Anwendung direkt auf eine Netzzugangsschnittstelle, wie etwa NetBIOS in MS-DOS, ist eine Optimierung dieser Anwendung hinsichtlich der Intensität der Kommunikation zwischen ihren Instanzen möglich. Dieser Vorteil fällt vor allem bei den relativ langsamen Netzen, wie sie in der DDR zur Verfügung stehen (SCOM), ins Gewicht, ist aber auch im Zusammenhang mit der Übertragung in Weitverkehrs- oder ISDN-Netzen von prinzipieller Bedeutung /1/.

Beispielsweise soll von einem verteilten Datenbank-System an einer Arbeitsstation eine Anfrage etwa in der in dBase üblichen Form LIST FOR ... gestartet werden. Diese Anfrage kann in Form einer Nachricht einem Datenbank-Verwaltungssystem (DBMS) zugesandt werden, das auf einem anderen Rechner im Netz, dem Datenbank-Server, installiert ist. Der Datenbank-Server beantwortet diese Anfrage und schickt das Ergebnis an die Arbeitsstation zurück. Insgesamt gehen für diese Anfrage also zwei Nachrichten über das Netz.

Der Nachteil von verteilten Anwendungssystemen besteht darin, daß aufgrund des niedrigen funktionellen Niveaus der Netzzugangsschnittstelle eine große Anzahl von immer wieder vorkommenden Verwaltungsfunktionen für jedes verteilte Anwendungssystem neu erarbeitet werden muß.

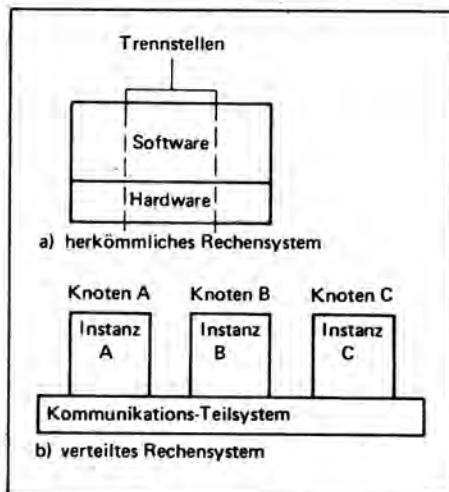


Abb. 2 Übergang von einem herkömmlichen Rechensystem zu einem verteilten (Rechen-) System

Beispielsweise können verschiedene Anwendungsprogramme über eine im Betriebssystem des jeweiligen Rechners vorhandene Netzzugangsschnittstelle auf entfernte Dienste zugreifen. Jedes Anwendungs-Diensterbringer-Paar hat dazu seine eigenen speziellen Schnittstellen und Protokolle. Für alle neu zu implementierenden Netzdienste müssen alle Probleme, wie Dienstadressierung, Nutzeranmeldung, Sicherung der Datenintegrität, Verwaltung nebenläufiger Zugriffe auf die Dienstleistung usw. neu gelöst werden.

Verteilung von Betriebssystem-Komponenten

Die Implementierung von Betriebssystemen erfolgt üblicherweise durch zwei Software-Schichten,

- den Systemdiensten und
- dem Betriebssystemkern.

Die *Schicht der Systemdienste* realisiert die Nutzerschnittstelle des Betriebssystems. Aufbauend auf die Basismechanismen eines Kerns werden dem Nutzer Dienstleistungen in Form von Systemrufen angeboten, wie etwa das Aktivieren eines Anwender-Prozesses, die Ausgabe eines Zeichens auf dem Bildschirm, das Schreiben in eine Datei, das Bereitstellen eines Datenpuffers usw. Letztendlich erfolgt der Zugriff des Nutzers auf die benötigten Betriebsmittel über diese Systemdienst-Schnittstelle. Unter DOS ist das beispielsweise die mit den Interrupts 21H bis 2FH gegebene Schnittstelle.

Der *Kern*, für den Nutzer meist nicht zugänglich, realisiert alle Grundmechanismen der Prozeß-, Geräte- und Spei-

cherverwaltung, wie etwa die Zuordnung eines Prozesses zu einem Prozessor, die Aufteilung des Speichers in logische Segmente, Geräte-Treiber usw. Im DOS bildet das ROMBios den Kern. Durch eine Verteilung von Betriebssystemkomponenten erhofft man sich u. a. nachfolgende Effekte:

- Schaffung einer hohen Fehlertoleranz und damit Steigerung der Verfügbarkeit und Zuverlässigkeit
- Modularer Aufbau aus zum Teil billigen Komponenten (Mikroprozessoren) und damit Erweiterbarkeit hinsichtlich – neuer Systemdienste
- neuer Systemressourcen und
- der Leistungsfähigkeit
- Steigerung der Leistungsfähigkeit durch Parallelität und bessere Ressourcenauslastung.

Für die Anwendungen wird die Verteilung von Betriebssystemkomponenten im allgemeinen nicht sichtbar (Transparenz).

Je nachdem, welche Komponenten eines Betriebssystems verteilt werden, lassen sich zwei Stufen der Verteilung von Betriebssystem-Komponenten unterscheiden,

- Netzbetriebssysteme und
- verteilte Betriebssysteme.

Netzbetriebssysteme

In Netzbetriebssystemen werden Teile der Systemdienstschicht verteilt – und zwar die Teile zur Verwaltung des Dateisystems und der Rechnerperipherie. Die Implementierung von Netzbetriebssystemen erfolgt günstigerweise als Erweiterung bestehender (Standard-) Betriebssysteme.

Netzbetriebssysteme (NOS: Network Operating Systems) sind Erweiterungen lokaler Betriebssysteme, die aufbauend auf ein Kommunikations-Teilsystem den Zugriff auf im Netzwerk entfernt erbrachte, von mehreren Nutzern im Netz geteilt nutzbare Systemdienstleistungen erlauben.

Auf entfernte und lokale Ressourcen wird in exakt derselben Weise zugegriffen, ohne daß zum Zeitpunkt des Zugriffs ein physischer Ort des Diensterbringers angegeben wird (Ortstransparenz).

Die Systemdienst-Schnittstelle wird üblicherweise um netzwerkspezifische Dienste und Funktionen, insbesondere um direkte Kommunikations-Dienste und Funktionen zur Unterstützung der durch die Ressourcenteilung entstandenen Mehrbenutzerumgebung erweitert. So wurde beispielsweise beim Übergang von der Version 2.11 zur Version 3.0 im DOS solch eine Erweiterung vorgenommen.

Zur Erzeugung der Ortstransparenz muß es auf Seiten des Dienstanwenders eine Einrichtung geben, die entscheidet, ob ein Systemdienst vom lokalen System oder über das Netz erbracht wird. Eine derartige Einrichtung soll hier als *Interzeptor* bezeichnet werden. Andere, bei konkreten Produkten dazu synonym verwendete Begriffe sind Redirector, Requestor oder Shell. Der Interzeptor greift auf Informationen zurück, denen er entnehmen kann, welche Dienste wo realisiert werden.

Die diensterbringende Instanz wird als *Server* bezeichnet. Server können als Anwendung eines Standardbetriebssystems wie etwa DOS oder UNIX oder als eigenständige Systeme unter Steuerung eines eigenen, darauf spezialisierten Betriebssystems implementiert sein. Anwendungen, die das Netzbetriebssystem nutzen, können die üblichen Anwendungen des ursprünglichen, nicht erweiterten Betriebssystems sein oder spezielle, netzwerkfähige Anwendungen.

Die *netzwerkfähigen Anwendungen* gestatten eine geteilte und quasiparallele Nutzung von im Netz zentral oder verteilt verwalteten Ressourcen. Das Anwendungsprogramm selbst besteht nicht aus im Netzwerk verteilten Instanzen, sondern läuft als ganzes auf mehreren Knoten im Netz gleichzeitig.

Beispielsweise kann ein Datenbanksystem für Personalcomputer, wie z. B. dBASE III Plus, zur gleichen Zeit auf mehreren PC im Netz laufen und die zentral verwalteten Datenbank-Dateien geteilt nutzen. Dieses Konzept ist analog dem klassischen Time-Sharing-Konzept mit Terminals, mit dem wesentlichen Unterschied, daß sämtliche Verarbeitung lokal stattfindet.

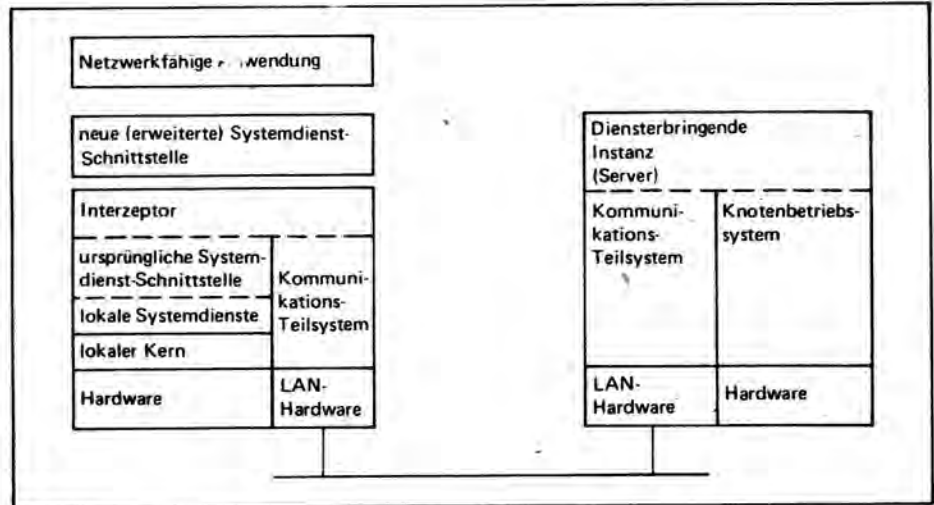


Abb. 3 faßt die Strukturelemente eines Netzbetriebssystems zusammen.

Indem die im Netz verteilten Dienste dem Nutzer über die übliche Systemdienst-Schnittstelle, d. h. wie die lokalen Dienste bereitgestellt werden, stellen die Leistungen und Ressourcen des Netzes nur eine Erweiterung der vorhandenen Umgebung dar. Die Netzwerk-Leistungen werden also dem Betriebssystem angepaßt und nicht umgekehrt. Dem Benutzer erscheinen die Netzumgebung, ihre Dienste und Ressourcen als eine Einheit. Für das Betriebssystem etablierte Standardsoftware bleibt daher weiterhin nutzbar, jetzt sogar mit der Möglichkeit, auf die im Netz verteilten Ressourcen zuzugreifen.

Während verteilte Anwendungen in bezug auf die Kommunikationsintensität optimal implementierbar sind, aber auf Grund der niedrig gelegenen Netzzugangsschnittstelle eine geringe Standardisierung von Teilen ihrer Funktionalität aufweisen, ist es bei Netzbetriebssystemen genau umgekehrt: Sie sind kommunikationsintensiv, weisen aber eine durch das jeweilige Betriebssystem gegebenen hohen Grad an Standardisierung von Funktionen auf, so daß sie der Anwendungsprogrammierer als gegeben ansehen kann. Weiterhin ist für den Anwendungsprogrammierer eines NOS das Netz bereits nicht mehr sichtbar. Der Programmierer einer verteilten Anwendung muß sich aber noch mit netzspezifischen Problemen auseinandersetzen.

Die oben als Beispiel angeführte Datenbankabfrage LIST FOR ... würde in einem Netzbetriebssystem etwa so ablaufen: Zunächst wird ein Ruf *Openfile* an den Datei-Server gesandt, danach so-

Abb. 3 Strukturelemente eines Netzbetriebssystems

lange *ReadFile*-Rufe, bis die entsprechenden Daten gefunden wurden, worauf ein *CloseFile*-Ruf an den Server geht. Es kann aber auch durchaus sein, daß in dieser Datei die gewünschten Daten nicht vorhanden sind, so daß die gesamte Datei völlig umsonst über das Netz gelesen wurde.

Die im NOS gegebene Funktionalität hat den Anspruch auf Universalität, da das Netzbetriebssystem meistens die Erweiterung eines Standard- (Universal-) Betriebssystems ist. Damit ist aber durchaus nicht jeder Anwendung geholfen.

Als Basis für netzwerkfähige Anwendungen sind Netzbetriebssysteme nur dann wirklich geeignet, wenn die Antwortzeit bei der Nutzung eines entfernt im Netz verwalteten Betriebsmittels nicht wesentlich größer ist, als die für einen lokalen. Diese Eigenschaft wird als *Leistungstransparenz* bezeichnet.

Mit fortschreitender Technologieentwicklung (schnelle Netze, intelligente LAN-Controller mit hoher Rechenleistung zur Verarbeitung der Kommunikationsprotokolle und eine hohe Leistungsfähigkeit der Server) wird diese Forderung immer besser erfüllt.

Entgegen dem Einsatzfall als Basis von netzwerkfähigen Anwendungen möchte man mitunter einfach nur teure oder schlecht beschaffbare Geräte für mehrere Nutzer zugänglich machen – und zwar in der für die Nutzer üblichen Systemumgebung. Hier liegt das Einsatzfeld von Billig-Systemen, wie sie zahlreich international angeboten werden,

Verteilte Verarbeitung – Prinzipien und Modelle

	Netztransparenz		Netzbelastung	Universalität des Funktionsumfangs	Standardisierung von Teilen des Funktionsumfangs
	für Anwendungsprogrammierer	für Nutzer			
verteilte Anwendungen	nein	möglich	gering möglich	Funktionsumfang anwendungsspezifisch	gering
Netzbetriebssysteme	ja	ja	noch	Funktionsumfang universell (betriebsspezifisch)	noch (durch Betriebssystem gegeben)

etwa auf der Basis einfacher V.24-Verbindungen.

In der Tab. werden die Vor- und Nachteile von Netzbetriebssystemen im Vergleich zu den verteilten Anwendungen noch einmal zusammengefaßt.

Netzbetriebssysteme sind vor allem für die Arbeit in PC-Arbeitsgruppen auf der Basis abgeschlossener Teilnetze, z. B. auf Abteilungsebene, geeignet. Dabei können je nach Leistungsfähigkeit 5 ... 30 Arbeitsstationen an einem Server angeschlossen sein.

Für die Kommunikation über das Teilnetz hinaus, etwa zur Nutzung betriebsweit verteilt verwalteter Datenbestände, erweisen sich verteilte Anwendungen jedoch als günstiger.

Für CP/M gab es das Netzbetriebssystem CP/Net /2/, das seiner Leistungsfähigkeit nach den Billig-Systemen zuzuordnen wäre. Es hat international kaum Verbreitung gefunden. Das in /3/ vorgestellte MicroNET-80 ist ebenfalls unter CP/M lauffähig, genauso wie das SCPNet vom VEB Kombinat Robotron /4/.

Für MS-DOS gibt es eine Reihe Netzbetriebssysteme. Zu nennen wären MS-NET /5/ von Microsoft und das darauf beruhende IBM-PC-Network-Program (IBM-PCNWP) /6/ der IBM und das System NetWare /7, 8/ der Firma Novell. Letzteres erfüllt die Forderung nach Leistungstransparenz.

Das Netzbetriebssystem für OS/2 aus dem Hause Microsoft wird der OS/2-LAN-Manager sein. Novell hat angekündigt, daß es das OS/2 ebenfalls unterstützen will.

Unter UNIX existieren Netzbetriebssysteme für Rechner der Workstation-Klasse, so z. B. das Network File System (NFS) /10/ der SUN-Microsystems oder die Network Computing Architecture (NCA) /11/ für APOLLO-Workstations.

Für Netzbetriebssysteme unter MS-DOS gibt es zahlreiche netzwerkfähige Anwendungen, wie etwa Dienst- und Hilfsprogramme, Datenbanksysteme (dBASE III Plus, dBASE TV), inte-

grierte Pakete (OPEN ACCESS) und Spezialpakete /12/.

Verteilte Betriebssysteme

Während bei Netzbetriebssystemen nur Teile der Systemdienste-Schicht eines Betriebssystems verteilt sind, ist es bei verteilten Betriebssystemen auch der Betriebssystem-Kern.

Ein verteiltes Betriebssystem (DOS: Distributed Operating System) verhält sich nach außen wie ein einheitliches, zentralisiertes System. Die einzelnen Rechner im Kommunikationssystem sind also nicht mehr selbständig, sondern Teil eines integrierten Gesamtsystems. Verteilte Betriebssysteme sind daher auch nicht als Erweiterung bestehender Betriebssysteme zu implementieren. Um die Kompatibilität zu solchen zu sichern, emuliert man daher deren äußeres Verhalten (z. B. UNIX-System V).

Der Nutzer weiß nicht mehr, wo beliebige von ihm benutzte Ressourcen residieren. Die Zuordnung der aktiven Prozesse erfolgt dynamisch auf die verteilten Prozessoren im Netz. Ebenso wird die physische Speicherung von Dateien auf im Netz verteilte Laufwerke automatisch durch das Betriebssystem in Abhängigkeit etwa von der Belegung der einzelnen Laufwerke vorgenommen. Bei Ausfall einer Teilkomponente können die gleichen Anwendungen wie sonst, jetzt aber unter geringerer Gesamtleistungsfähigkeit ablaufen.

Verteilte Betriebssysteme sind gegenwärtig international ein aktuelles Forschungsgebiet.

Ihre kommerzielle Bedeutung erscheint dem Autor als noch nicht umfassend nachgewiesen. Neben noch ungeklärten konzeptionellen Fragen liegt das Hauptproblem in der durch die intensive Kommunikation im verteilten Kern des Betriebssystems bedingte, noch unzulängliche Leistungsfähigkeit. Mit der preiswerten Verfügbarkeit von Hochleistungskommunikationssystemen, wie etwa das FDDI /13/, könnten sich verteilte Betriebssysteme als ökonomi-

Tab. Vergleich verteilter Anwendungen mit Netzbetriebssystemen

schere und flexiblere Alternative zu Hochleistungs-Rechnersystemen erweisen. Ein weiteres Einsatzgebiet wird in Bereichen gesehen, wo es auf extrem hohe Zuverlässigkeit ankommt, wie etwa in der Raumfahrttechnik oder im Militärwesen. Durch die hohe Fehlertoleranz verteilter Betriebssysteme wird bei Ausfall einer Systemkomponente nicht das gesamte System arbeitsunfähig, sondern es kann mit verminderter Gesamtleistungsfähigkeit seine Arbeit fortsetzen /14/.

Es sei darauf hingewiesen, daß die Begriffe Netzbetriebssystem (Network Operating System) und verteilte Betriebssystem (Distributed Operating System) in der Literatur oft ohne Unterscheidung angewandt werden.

Der Übergang von den Netzbetriebssystemen zu den verteilten Betriebssystemen ist fließend. So zeichnen sich sowohl beim LAN-Manager für das OS/2-Betriebssystem als auch bei der NCA bereits Eigenschaften ab, die auf die verteilten Betriebssysteme verweisen. Das betrifft z. B. die Möglichkeit beider Systeme, bestimmte Programmanteile eines laufenden Programms parallel auf anderen, gerade freien Prozessoren ablaufen zu lassen (Prozeß-Migration).

Zusammenfassung

Netzbetriebssysteme haben sich international trotz ihrer konzeptionell bedingten Kommunikations-Intensität sowohl bei PC-Netzen, als auch im Bereich der Graphic-Workstation eine große Marktrelevanz erobert. Das liegt vor allem daran, daß durch die international relativ preiswerte Verfügbarkeit von LAN z. B. der ETHERNET-Klasse, diese Kommunikations-Intensität als Nachteil nicht mehr so stark ins Gewicht fällt. Damit kommen die Vorteile eines durch das Basisbetriebssystem gegebenen hohen Standardisierungsgrades der Grundfunktionen verteilter Systeme voll zur Geltung. Verteilte Anwendungs-

NETBIOS – standardisierter Netzzugang für PC-Technik

Uwe Schulze, Wilhelm-Pieck-Universität Rostock
Institut für sozialistische Wirtschaftsführung, Informatik-Labor

systeme werden aber insbesondere bei der Nutzung von Übertragungskanälen mit geringem Durchsatz weiterhin ihre Bedeutung behalten. Insbesondere bei Systemen höherer Leistungsklasse, wie z. B. die Graphik-Workstations, lassen sich in der Architektur von Netzbetriebssystemen Elemente von verteilten Betriebssystemen erkennen.

Literatur

- /1/ Göde, B.: Die Bedeutung der Prozessallokation in verteilten LAN-Applikationen bei begrenzter Übertragungskapazität. In dieser Ausgabe
- /2/ Digital Research: CP/Net Users Guide. Pacific Grove 1980 (Firmenschrift)
- /3/ Schulz, U. u. a.: MicroNET-80 ein Netzbetriebssystem für 8-Bit Personalcomputer. In dieser Ausgabe
- /4/ Richter, J. u. a.: ROLANETI – ein lokales Netz des VEB Kombinat Robotron. NTB 30 (1986) 5, S. 146–149
- /5/ Hurwicz, M.: MS-DOS 3.1 makes it easy to use IBM PC's on a network. Data Communications December 1985, S. 223–237
- /6/ IBM: IBM PC Network Program; User's Reference Manual, Technical Reference Manual. Firmenschrift der IBM, Corp.
- /7/ Novell: Grundlagen der Advanced NetWare 2.1. Firmenschrift der Novell GmbH.
- /8/ Stickler, R. u. a.: NetWare – Hochtechnologie bei PC-Netzen. In dieser Ausgabe
- /9/ Lippert, M.: NFS – der Industriestandard für Netzwerk-File-Systeme. unix/mail 5 (1987) 1, S. 35–39
- /10/ Apollo Domain: Das Konzept, mit dem Apollo Domain Rechnerwelten verändert hat. Firmenschrift der Apollo Computer GmbH.
- /11/ CHIP Spezial: Lokale Netze. Würzburg 1987
- /12/ Warren, C.: Fiberoptic networks shine new light on high-speed transfers. Computer Design June 15, 1987 S. 45–52
- /13/ Tanenbaum, A. S. u. a.: Distributes Operating Systems. ACM Computing Surveys Vol. 17 No. 4, Dezember 1985, S. 419–470

Ein Grund für die geringe Verbreitung von Software in Netzumgebung lag in der Vergangenheit im Fehlen einer Schnittstelle, die den Applikationen den Netzzugang unabhängig von der konkreten Hardware, Netztopologie, und Zugriffsverfahren gestattet. Seit der Markteinführung 1985 gelten MS-DOS 3.1 und NETBIOS als Softwarestandard für die Nutzung von PC-Technik in lokalen Netzen.

Von Bedeutung ist die Beziehung zwischen NETBIOS und DOS ab Version 3.1. Mechanismen zur Sicherung der Datenkonsistenz (Locking-Mechanismen) können nicht nur durch eine verteilte Anwendung realisiert werden, sondern müssen auf Betriebssystemniveau zur Verfügung gestellt werden. Das geschieht in DOS über den SHARE-Befehl.

NETBIOS definiert dabei eine Schnittstelle zwischen der konkreten LAN-Hard- und Firmware (i. a. auf Transport- oder Linkebene) auf der einen und Betriebssystem oder Applikation auf der anderen Seite.

NETBIOS bietet die Dienste der unterliegenden fünf Schichten in Form von aufrufbaren Kommandos an. Nach /2/ realisiert NETBIOS die höchstmögliche Netzschnittstelle für anspruchsvolle verteilte Applikationen. Damit ist völlige Hardware-Unabhängigkeit gewährleistet. Ursprünglich auf dem IBM-PC-Network-Adapter und später auf dem IBM-Token-Ring-Network-PC-Adapter implementiert, gibt es heute NETBIOS für alle wichtigen Netzstandards. Im allgemeinen setzt NETBIOS direkt auf der Link-Schicht unter Nutzung der dort verfügbaren Datagram-Dienste auf und realisiert so die Schichten 3 bis 5 im Referenzmodell.

Auf Sitzungsniveau sind vor allem vier Aufgaben zu realisieren:

- Namensverwaltung (Node-Naming)
- Aufbau und Verwaltung virtueller (logischer) Verbindungen
- zuverlässige Datenübertragung
- Netzdiagnose (Status) und Wartungsfunktionen.

Die Schnittstellengestaltung des Betriebssystems MS-DOS gestattet den Zugang zu den Systemdiensten auf

BIOS- und BDOS-Ebene. Ebenso ist NETBIOS direkt unter Interrupt 5Ch aufrufbar oder auf DOS-Ebene mit Interrupt 2Ah. Nur fünf Prozent der Anwendungen sollen – hauptsächlich für Gateways – den direkten Zugriff nutzen /2/.

Durch Interrupt 2Ah werden neben dem Durchreichen des NETBIOS-Aufrufs noch folgende Funktionen zur Verfügung gestellt:

- Überprüfen der Installation von NETBIOS
- I/O – Überprüfen
- Ermitteln von Netz-Ressourcen-Informationen.

Für die gemeinsame Nutzung von Datenbeständen finden die ab DOS 3.1 unter Sammelinterrupt 21h angebotenen folgenden Funktionen Verwendung:

- Erweiterte Open-Funktion (3Dh)
- I/O-Kontrolle (44h)
- Ausführen eines Programms (4Bh)
- Erweiterten Fehlermodus einstellen (59h)
- Anlegen temporäre Datei (5Ah)
- Eröffnen neue Datei (5Bh)
- Sperren/Entsperren von Datensätzen (5Ch)
- Geräteverwaltung im Netz (5Eh und 5Fh).

Einige Hilfsfunktionen – besonders im Zusammenhang mit den Netzbetriebssystemen – sind auch noch im Multiplex-Interrupt 2Fh zu finden.

NETBIOS unterstützt jeweils zwei LAN-Adapter gleichzeitig, zwischen denen beim Funktionsaufruf ausgewählt werden kann. Damit steht ein einfacher Mechanismus für die Verbindung verschiedener Netze zur Verfügung.

Funktionalität

Die Vermittlung der Parameter an NETBIOS erfolgt über einen Zeiger, der auf den Network Control Block (NCB) weist, in dem alle funktionspezifischen Angaben enthalten sind.

Die NETBIOS-Funktionen lassen sich in vier Gruppen unterteilen /1/:

- Allgemeine Funktionen (General)
- Namen-Dienste (Name Support)
- Verbindungsorientierte Dienste (Session Support)
- Paket-Dienste (Datagram Support).

Länge (in Byte)	Funktion	Bedeutung
1	COMMAND	Befehlscode
1	RETCODE	Fehlercode
1	LSN	Sitzungszähler
1	NUM	Nummer des Netzwerknamens
4	BUFFER	Pufferadresse der zu übertragenden Daten
2	LENGTH	Pufferlänge
16	CALLNAME	Name des Zielknotens
16	NAME	Name des lokalen Knotens
1	RTO	Empfangs-Timeout in halben Sekunden
1	STO	Sende-Timeout in halben Sekunden
4	POST	Adresse einer nutzerdefinierten Routine, die nach NETBIOS ausgeführt wird
1	LANA	LAN-Adaptornummer (0 oder 1)
1	CMDCMPLT	Final-Rückkehrcode
14	RESERVE	reservierter Bereich

Tab. 1 Aufbau des Network Control Block

Allgemeine Funktionen

RESET

Initialisieren des LAN-Adapters; Ferner kann die Anzahl der gleichzeitig offenen Verbindungen (Sessions) und die Anzahl der Network-Control-Blocks festgelegt werden. Möglich sind jeweils 1 bis 32, Standard ist 6 bzw. 12.

CANCEL

Abbrechen eines Kommandos

ADAPTER STATUS

Ermittlung des Status des lokalen oder eines entfernten Netz-Adapters.

Namen – Dienste

Verbindungen werden über Namen (16 ASCII-Zeichen), statt der sonst üblichen Nummern abgewickelt. Damit sind die Applikationen unabhängig von den Konventionen der Transportdienste. Ferner ist es möglich, daß sich jeder Prozeß unter einem anderen Namen bei seinem NETBIOS anmeldet. Sofern mehrere Prozesse gleichzeitig auf einer Station laufen, bestehen Verbindungen nicht auf dem Niveau Station-Station, sondern auf dem von Prozeß zu Prozeß.

ADD_NAME

Eintragen eines Namens in die Namens-Tabelle. Der Name kann nicht doppelt verwendet werden.

ADD_GROUP_NAME

Eintragen eines Namens in die Namens-Tabelle. Andere Prozesse können sich unter dem gleichen Namen anmelden.

DELETE_NAME

Löschen eines Namens aus der Namens-Tabelle.

Jeder Knoten verwaltet eine Namens-Tabelle mit bis zu 16 Namen. Dort werden jeweils nur die lokalen Namen gespeichert. Auf anderen Knoten angemeldete Prozesse werden per Rundruf ermittelt. Der Namensdienst ist also dezentral organisiert.

Verbindungsorientierte Dienste

Mit verbindungsorientierten Diensten werden logische Verbindungen zwischen Prozessen geöffnet, über mehrere Sende- und Empfangsaktionen hinweg erhalten und, wenn nicht mehr benötigt, explizit wieder geschlossen.

CALL

Anforderung eines Verbindungsaufbaus;

LISTEN

Warten auf Verbindungsanforderung und Quittieren;

HANG_UP

Beenden einer logischen Verbindung;

SEND

Senden eines Datenpaketes zu einem

spezifizierten Prozeß, zu dem eine logische Verbindung besteht;

CHAIN_SEND

Senden zweier verketteter Datenpakete wie bei SEND;

RECEIVE

Empfangen eines Datenpaketes von einem spezifizierten Prozeß, zu dem eine logische Verbindung besteht;

RECEIVE_ANY

Empfangen eines Datenpaketes von einer beliebigen Station, zu der eine logische Verbindung besteht;

SESSION STATUS

Ermittlung von Informationen über alle Verbindungen, die zum spezifizierten Namen gehören. Es ist auch möglich, alle eingetragenen Namen der lokalen Station zu erhalten.

Alle bestehenden logischen Verbindungen werden in einer Tabelle (Session Table) eingetragen. Dort werden neben den beiden beteiligten Prozeßnamen ein Sende- und Empfangs-Timeout (STO,RTO) sowie statistische Werte zur jeweiligen Verbindung abgelegt.

Paket-Dienste

SEND_DATAGRAM

Senden eines Datenpaketes zu einem spezifizierten oder einer Gruppe von Prozessen;

SEND_BROADCAST_DATAGRAM

Senden eines Datenpaketes an alle Stationen;

RECEIVE_DATAGRAM

Empfangen eines Datenpaketes, das an den empfangenen Prozeß gerichtet ist;

RECEIVE_BROADCAST_DATAGRAM

Empfangen eines Paketes, das an alle Stationen gerichtet ist.

Parallelarbeit

NETBIOS unterstützt den nebenläufigen Betrieb von Applikations- und Nebenprozessen. Der NETBIOS-Nutzer hat die Wahl zwischen blockierender synchroner sowie nichtblockierend asynchroner Kommunikation. Bei Nutzung der WAIT-Option (synchron) wartet der Anwenderprozeß, bis die geru-

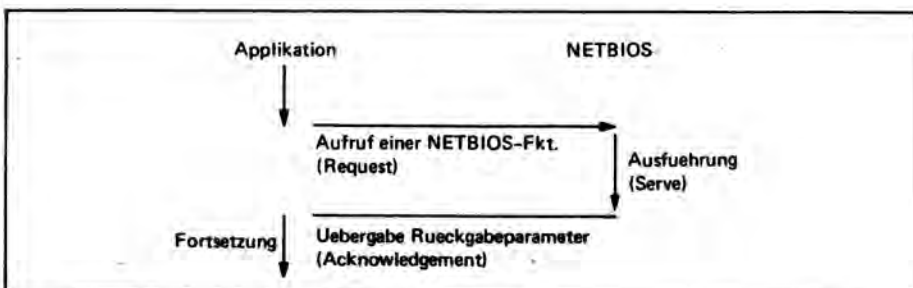


Abb. 1 Synchroner NETBIOS-Prozeß-kommunikation

Verteilte Verarbeitung – Prinzipien und Modelle

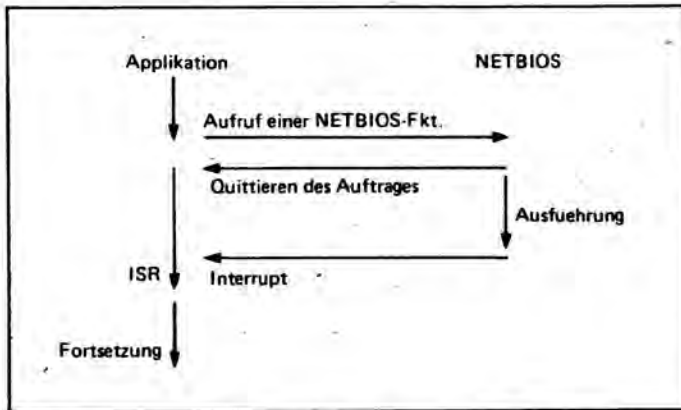


Abb. 2 Asynchrone NETBIOS-Prozesskommunikation mit aktiver Quittung

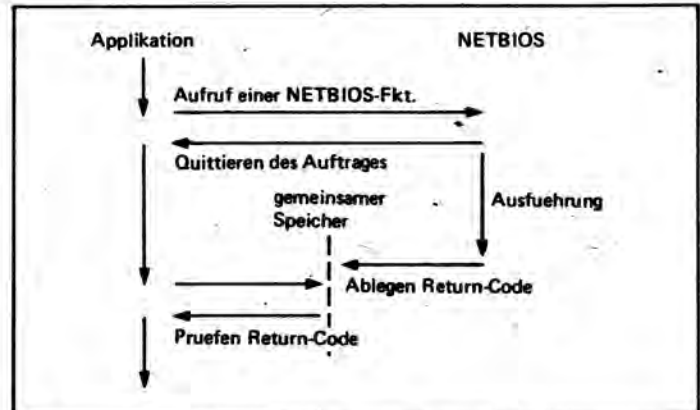


Abb. 3 Asynchrone NETBIOS-Prozesskommunikation mit passiver Quittung

fene Netzfunktion vollständig ausgeführt ist. Über den Rückkehrcode ist eine erfolgreiche bzw. fehlerhafte Ausführung zu unterscheiden. Abb. 1 zeigt die Prozesskommunikation.

Der Anwendungsprozeß bleibt für die Zeitdauer der Ausführung der Funktion blockiert. Auch bei Verwendung intelligenter Netzadapter kann diese Art der Nutzung von Netzdiensten günstig sein, nämlich dann, wenn das Anwenderprogramm sequentielle Abarbeitung erfordert. An dieser Stelle sei angemerkt, daß die den Netzfunktionen rufen ähnlichen Systemrufe im Betriebssystem stets einer Arbeit im WAIT-Modus entsprechen.

Die NO-WAIT-Option (asynchron) trägt den besonderen Bedingungen Rechnung, die durch die Kommunikation zwischen unabhängigen Arbeitsstationen (PC) gegeben sind, nämlich: Die Zeit für die Ausführung ist nicht determiniert. Ursachen dafür sind:

- unterschiedliche Netzbelastung
- Fehlererkennungs- und Beseitigungsmechanismen auf allen Ebenen (gegebenfalls wiederholtes Übertragen von Paketen)
- Notwendigkeit des Wartens, wenn eine Station besetzt (busy) ist.

Um den Anwendungsprozeß unabhängig von den zeitlichen Netzgegebenheiten zu gestalten, wird bei Nutzung der NO-WAIT-Option von NETBIOS nur die Annahme des Auftrages zur Ausführung einer Funktion quittiert. Die Steuerung kehrt anschließend an das Anwenderprogramm zurück, und die Ausführung der Funktion erfolgt unabhängig und zeitlich parallel. Das Anwenderprogramm muß aber noch die erfolgreiche Ausführung der Funktion prüfen und gegebenenfalls die Rückgabeparameter entgegennehmen. Hierfür stellt NET-

BIOS eine aktive und eine passive Variante zur Verfügung.

Bei der Quittung im aktiven Modus spezifiziert das Anwenderprogramm die Adresse einer Interrupt-Service-Routine, die nach Ausführung der Funktion von NETBIOS durch maskiertes Interrupt aktiviert wird. (Abb. 2)

Im passiven Modus trägt NETBIOS an eine festgelegte Stelle im übergebenen Parameterblock einen Rückkehrcode ein, wenn die Ausführung abgeschlossen ist. Es obliegt dem Anwenderprozeß, diesen zyklisch zu testen (Polling) und, wenn erforderlich, Fehlerbehandlungsmaßnahmen einzuleiten. Die Kopplung der Prozesse erfolgt hier also über einen gemeinsam genutzten Speicherbereich. Abb. 3 zeigt den Vorgang. Damit stehen den Anwendungsprozessen ausreichende Mittel zur Verfügung, die Kommunikation mit der Netzsoftware so zu gestalten, daß sie den Besonderheiten der Anwendung in Bezug auf das Zeitverhalten am besten entspricht. So ist ein Optimieren der Reihenfolge der Ausführung von Anforderungen aus der Warteschlange (Scheduling) ebenso möglich, wie das quasiparallele Abarbeiten mehrerer Anforderungen mittels Time-Sharing oder die Nutzung von Wartezeiten (z. B. Peripheriearbeit) zum Abarbeiten nachfolgender Anforderungen.

Zusätzlich hat der Anwendungsprozeß die Möglichkeit, noch nicht vollständig bearbeitete Anforderungen abzubrechen (Kommando CANCEL).

Knotenverwaltung im Netz

Die Kommunikation mit NETBIOS erfolgt nicht auf Ebene der Netzknotten, sondern auf Prozeßebene. In jedem Netzadapter werden 16 Prozesse ver-

waltet. Zur Identifikation werden Namen (16 Zeichen lang) verwendet. Mit diesen Namen wird also stets ein Prozeß angesprochen, keine Arbeitsstation! Damit ist für die Arbeit im Netz irrelevant, auf welchem Netzknotten ein Prozeß läuft, was eine ortstransparente Nutzung von Netzdiensten gestattet. Dafür muß gesichert werden, daß die Namen jeweils einmalig sind, was bedeutet, daß bei Aufnahme eines neuen Namens alle Knotten abgefragt werden müssen. Gleiches gilt für jegliche Datenübertragung. Da nicht bekannt ist, auf welcher Station der mit dem Zielnamen zu identifizierende Prozeß installiert ist, müssen alle Netzknotten daraufhin geprüft werden. Dies erfordert leistungsfähige (schnelle) Hardware und ist besonders bei Verbindung mehrerer Netze kompliziert zu realisieren.

Eine Alternative ist die im Netzbetriebssystem NETWARE von Novell praktizierte hierarchische Netzorganisation /4/. Dabei verwaltet nicht jede Station die Namen der lokalen Prozesse, sondern ein Namenserver alle Namen im Netz. Damit können alle Netzrufe direkt adressiert werden und müssen nicht als Rundruf verschickt werden. Die Nachteile liegen in der Abhängigkeit der Sicherheit des gesamten Netzes von der Masterstation, vor allem aber darin, daß eine ortstransparente Nutzung von Netzdiensten nicht möglich ist, da dem rufenden Prozeß bekannt sein muß, wo der dienstbringende Prozeß angemeldet ist.

Zusammenfassung

Mit der Nutzung von NETBIOS verbinden sich zwei Ziele:

- Vorhandene Standardsoftware kann über das Netzbetriebssystem IBM PC Network Program durch Arbeit mit vir-

Modell eines verteilten Datenbank-Betriebs-Systems (VDB)

Prof. Dr. Gerd Rosse, Wilhelm-Pieck-Universität Rostock
Institut für sozialistische Wirtschaftsführung, Informatik-Labor

tuellen Laufwerken für die Netzarbeit erschlossen werden. Alle Programme, die exakt die BDOS-Schnittstelle nutzen, lassen sich so verwenden. In neuen Versionen werden außerdem Funktionen für den gemeinsamen Datenzugriff zur Verfügung gestellt. Aufgrund der hohen Netzbelastung bei dieser Methode ist ein recht leistungsfähiges Netz Voraussetzung.

Der Programmierer erhält eine Hardware-unabhängige Schnittstelle. Eine Portierung wird damit leicht möglich.

Für das gerade vorgestellte neue Betriebssystem OS/2 ist bereits ein OS/2-LAN-Manager angekündigt. Durch Ausnutzung der Multitasking-Fähigkeit kann dann ein PC problemlos gleichzeitig Server (im Hintergrund) und Arbeitsstation sein. Die aus der UNIX-Welt übernommenen Systemfunktionen zur Datensicherung und Interprozeßkommunikation finden dabei Verwendung. An der Philosophie des Netzzuganges ändert sich aber nichts. Es wird ein OS/2-NETBIOS geben (natürlich ist im Protected Mode die Nutzung des Interrupts 5Ch nicht mehr möglich). Neben der beabsichtigten gleichzeitigen Nutzung von OS/2 und MS/DOS-Rechnern in einem Netz soll dann auch die Einbeziehung von UNIX möglich sein.

Am ISW der WPU ist derzeit eine NETBIOS-Implementation auf der Grundlage des lokalen Netzes SCOM-LAN der IHS für Seefahrt Warnemünde/Wustrow /5/ in Arbeit.

Literatur

- /1/ IBM – Technical Reference: Token Ring Network, PC Adapter Firmenschrift IBM Inc.
- /2/ Mike Hurwicz: MS-DOS makes it easy to use IBM-PC's on a network. Data Communications, Nov. 85
- /3/ Gregory Ennis: PC-Network Services for distributed System Design. IEEE Computer Society Press 1986 S. 155–160
- /4/ NOVELL: LAN-Operating System Report 1986. Novell Inc. 1987 S. 23–27
- /5/ Goede, B. u. a.: SCOM-LAN Benutzerhandbuch. Warnemünde, Sömmerda 1987

Die reale betriebliche DV-Welt macht derzeit einen zweiten Strukturwandel durch. Nachdem für relativ lange Zeit die zentrale DV der dominierende Strukturtyp war, kam es mit Eintritt der PC in die DV zu einer sich fast überschlagenden Distribution der informationsverarbeitenden Prozesse. Fast zum gleichen Zeitpunkt wurden die Nachteile der fehlenden informationellen Kopplungen der einzelnen Systeme untereinander offensichtlich. Die Rechnerkommunikation mit allen ihren Entwicklungsrichtungen sollte Abhilfe schaffen. Mit der Bereitstellung dieser Kommunikationsmöglichkeiten traten sowohl die praktischen als auch die theoretischen Probleme in der effektiven Beherrschung distributiver Systeme mit ihren notwendigerweise kooperierenden aber verteilten Prozessen zutage. Es fehlten theoretische Modelle und praktische Lösungen auf der Ebene der Applikationen, die den Fakt der Distribution als Baseeigenschaft der Systeme akzeptierten und durchgehend derartig berücksichtigten, daß gerade dieser Fakt auch positiv ausgenutzt wurde.

Aufgesetzte Komponenten, die eine sogenannte Netzwerkfähigkeit realisieren sollen, erschließen dem Nutzer zwar im Nachhinein die Netzressourcen, ändern aber nichts an der den Systemen zugrunde liegende Entwurfsphilosophie der Einplatzsysteme.

Die verteilten Systeme sind inzwischen ein Hauptgegenstand der Informatikforschung, was nicht zuletzt daran sichtbar wird, daß es schwerfällt, die dazu erscheinenden Veröffentlichungen zu verfolgen und die verschiedensten Modell- und Begriffsschöpfungen sinnvoll einzuordnen.

Das im Folgenden beschriebene Arbeitsmodell eines verteilten Prozeßsystems dient der Herstellung eines Verteilten-Datenbank-Betriebssystems

(VDB) und lehnt sich im wesentlichen an die in /1/ formulierten konzeptionellen Betrachtungen an, wobei wir insbesondere versucht haben, das Konzept nach oben hin in Richtung Applikation auszudehnen. Denn erst die eigentliche Applikation macht ein Konzept wir-

kungsvoll und sorgt durch ihre Rückkopplung für seine Weiterentwicklung. Das Informatik-Labor der WPU Rostock widmet sich in besonderem Maße den Decision-Support-Systems (DSS). Als notwendige Voraussetzung wirkungsvoller DSS ist eine flexible und umfassende Informationsversorgung anzusehen, woraus das Interesse an dem Problem des Datenmanagements in verteilten Systemen resultiert.

Zielstellung

Das hier formulierte Modell eines VDB versucht, mit dem Konzept der kooperierenden Prozesse den Fakt der DV-Distribution durchgängig und effektiv zu berücksichtigen.

Das VDB ist wiederum die unterste Schicht eines modular strukturierten Decision-Support-Systems, das alle Dienste der darunter liegenden Schichten nutzen kann.

Das Datenmodell des VDB (erweitertes Relationenmodell) ist so gestaltet, daß es dem Typ der Non-Standard-DBS entspricht, womit es durch die Wahl der zur Verfügung stehenden Datentypen und Datenverwaltungsroutinen in der Lage ist, neben wie üblich strukturierten Daten auch Texte (Dokumente), Programme und Wissensbasen (Regeln/Fakten) in einer Datenbasis zu verwalten. Das Datenbeschreibungssystem (DDS) ist ebenfalls so angelegt, daß sowohl das VDB als auch das DSS die jeweils notwendigen syntaktischen oder semantischen Informationen für sich separieren kann. Diese Komponenten sind bereits implementiert und werden zur Zeit in das obige VDB integriert.

Umfassendes Ziel dieser Arbeiten ist es, die Voraussetzungen für kooperierende wissensbasierte Systeme zu schaffen.

Das Prozeßmodell als Lösungsansatz

Wichtig erscheint uns (und dies nicht nur für den hier vorliegenden Gegenstand des VDB), DV-Aufgaben in verteilten Systemen modellmäßig als eine Folge kooperierender und partiell parallelisierbarer Prozesse aufzufassen. Alle darauf aufbauenden Applikationen, z. B.

ein Management-Information-System (MIS) auf der Basis eines Methodenbanksystems, lassen sich mit derselben Modellvorstellung über kooperierende Prozesse entwerfen und implementieren. Das Entwurfskonzept des VDB basiert auf folgender einheitlicher Modellvorstellung:

Ein verteiltes System besteht aus zeitlich parallel arbeitenden DV-Prozessen, die über logische Botschaftenkanäle untereinander und mit der Umwelt kommunizieren. Prozesse können zu Teilsystemen (Aktionen) zusammengefaßt werden, die dann wiederum Prozesse eines übergeordneten Systems sein können. Dementsprechend strukturiert sich die zu erstellende Software in Systeme und Prozesse, wobei die Prozesse (technologisch gesehen) aus Modulen bestehen. Ein Prozeß ist eine Sequenz von Berechnungen, durch die eine in sich abgeschlossene Funktion der Applikation realisierbar ist. Er besitzt einen systemweit eindeutigen Namen. Dieser Prozeß ist in seiner Einmaligkeit ein abstrakter Prozeß (AP), der durch seinen Aufruf zu einem konkreten Prozeß (P) wird, und durch einen definierten Entstehungs- und Terminierungszeitpunkt gekennzeichnet ist.

Prozesse sind entkoppelt, wenn während des Zeitintervalls ihrer konkreten Existenz die von den Prozessen genutzten Datenmengen paarweise disjunkt sind.

In einem verteilten System können deshalb auch gleichzeitig mehrere Ausführungen des gleichen abstrakten Prozesses aktiviert sein, sofern sie disjunkte Datenmengen bearbeiten. Sie sind durch eine systeminterne laufende Nummer eindeutig identifizierbar.

Prozesse sind dagegen gekoppelt, wenn sich die jeweils benutzten Datenbereiche während der gemeinsamen Existenz der beiden Prozesse überlappen.

In dem hier abgeleiteten Modell eines verteilten Datenbankbetriebsystems (VDB) werden grundsätzlich entkoppelte Prozesse zugrunde gelegt.

Eine zeitliche Folge von Prozessen kann zu einer Aktion zusammengefaßt werden, die dann eine komplexe Funktion realisiert. Aktionen sind damit die ei-

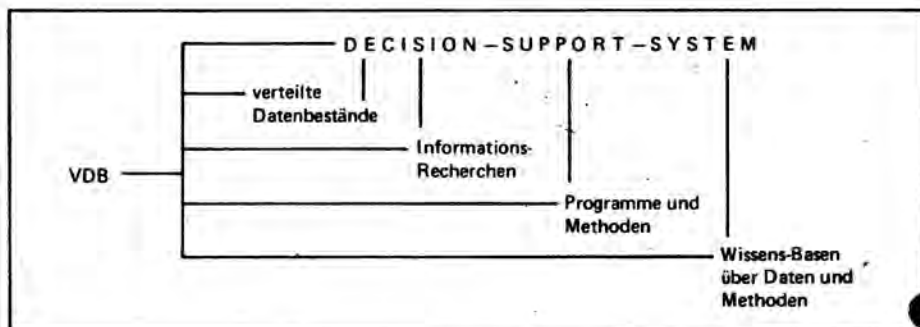


Abb. 1 Modular strukturiertes Decision-Support-System

gentlichen atomaren Elemente der Applikationen. Derartige Prozeßfolgen heißen kooperierende Prozesse. Kooperierende Prozesse sind dadurch gekennzeichnet, daß für sie eine zeitliche Abfolge festliegt, die entweder durch die Übergabe von Ergebnisdaten eines Prozesses zu einem Folgeprozeß oder durch Integritätsbedingungen determiniert ist. Deshalb haben die hier betrachteten Aktionen Ähnlichkeit mit den in der Literatur behandelten gekoppelten Prozessen /1/ mit dem Unterschied, daß im vorliegenden Modell die logische Prozeßkopplung auf einer höheren Abstraktionsbene erfolgt. Diese logische Kopplung wird durch das zeitliche Nacheinander der Prozesse und durch die zeitlich nicht überlappende Übergabe von Daten (Prozeßentkopplung) realisiert.

In bezug auf die Komplexität der für einen Prozeß erforderlichen Daten werden die Prozesse in drei Kategorien eingeteilt:

1. Der Prozeß wirkt auf einen Datensatz
2. Der Prozeß wirkt auf eine Datei
3. Der Prozeß wirkt auf die Datenbasis.

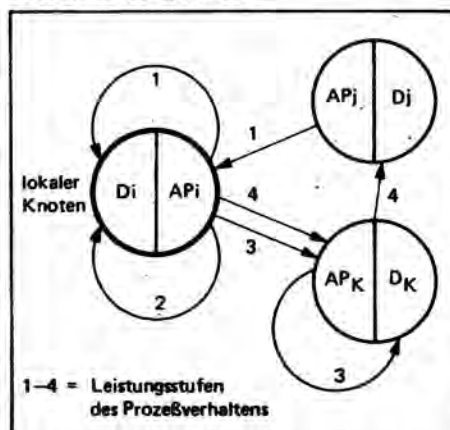
Diese Kategorisierung bildet die Basis der notwendigen konsistenzhaltenden Maßnahmen des VDB.

Auch in diesem Modell wird deshalb das bekannte Semaphorkonzept /2/ als abstrakter Sperrmechanismus unterlegt, wobei sich die Voraussetzung der Unteilbarkeit von Operationen aus dem Semaphorkonzept hier mit Einschränkung auf die Aktion bezieht. Diese Unteilbarkeit der Aktion gilt nur für den Sperrmechanismus. Ansonsten ist der Prozeß die atomare Operation. Diese Modellvorstellung hat den Vorteil, daß sich die

Prozeßsynchronisation in einem verteilten System einfacher beherrschen läßt. Aktionen erhalten deshalb die höchste Sperrkategorie der beteiligten Prozesse. Dieses Konzept erlaubt es dem steuernden verteilten Datenbank-Betriebs-System, folgenden Anforderungen gerecht zu werden:

- Die Synchronisation mehrerer im verteilten System gleichzeitig existierender Prozesse ist durch die Forderung der Entkopplung eindeutig geregelt. Synchronisation ist nur noch innerhalb von Aktionen notwendig.
- Die Sicherung der Integrität ist durch das Aktionenkonzept übersichtlicher geworden.
- Diejenigen Prozeßverklümmungen, die ihre Ursache im Sperrmechanismus haben, sind durch das Aktionenkonzept unmöglich geworden.

Abb. 3 Prozeßverhalten



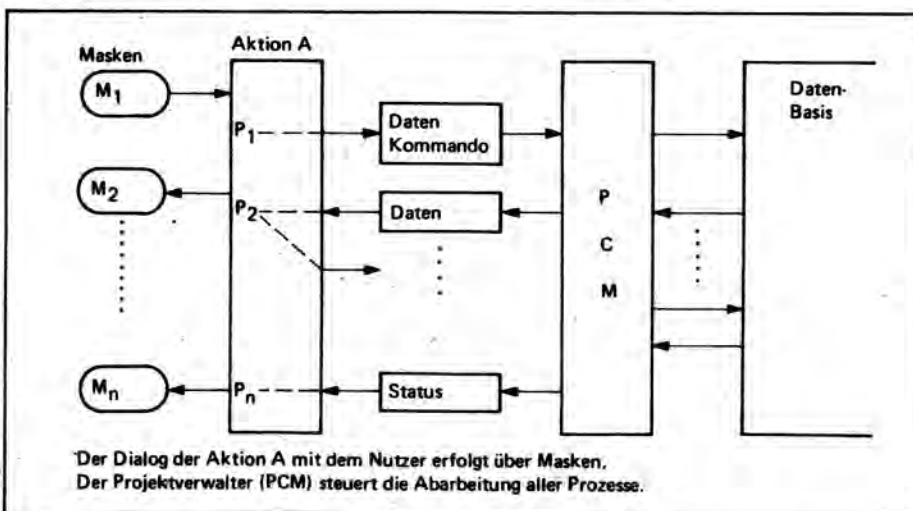
1. Kommunikation
 - 1. User-Check (Knotenzugang in/out)
 - Remote Procedure Call (RPC)
 - externes Time-Out Handling
2. PCM (Process Control/Management)
 - Prozeßverwaltung
 - 2. User-Check (Prozeßzugriff)
 - Verwaltung aller am lokalen Knoten anstehenden Prozesse (aus dem Netz und dem lokalen Knoten)
 - Verwaltung aller ans Netz abgehenden Prozesse
 - Prozeß-Logfile
 - Übergang zwischen
Netzwerkorientierten Aktivitäten
Datenbankorientierten Aktivitäten
3. VDB-Kern
 - 3. User-Check (DB-Zugriff)
 - DDS_COPY
 - P_CALL
 - Log-File
4. Permanenter DB-Block
 - 4. User-Check (Datei-Zugriff)
 - enthält alle residenten Informationen des DDS
5. Temporärer DB-Block
 - 5. User-Check (Satz-Zugriff)
 - enthält alle für den jeweiligen DB-Prozeß erforderlichen Datensätze

Abb. 2 Komponenten des VDB

Supplementation des VDB-Modells

Ein verteiltes System besteht aus den folgenden Komponenten, die das Mo-

Abb. 4 Prozeßverwalter



dell jeweils in einer Ebene widerspiegeln müssen und durch das VDB zu steuern sind:

- verteilte Datenbasis (VD)
- verteilte Menge abstrakter Prozesse (AP).

Um aus der Sicht einer beliebigen Applikation vollständige Transparenz zu sichern, steht (natürlich abhängig von dem gesetzten Zugriffsschutz) sowohl die gesamte Menge von Daten als auch die Menge von implementierten abstrakten Prozessen zur Verfügung. Dabei ist es unerheblich, ob ein bestimmter AP auf dem eigenen Knoten implementiert ist, oder ob es sich um einen ganz spezifischen Prozeß im Netz (z. B. Printer) handelt.

Dies wird dadurch erreicht, daß jedem Knoten ein Verzeichnis aller AP bekannt ist, und die parametrisierte Prozeßbezeichnung dem VDB erlaubt, die notwendigen Steuerinformationen (gegebenenfalls) der Netzwerksoftware zu übergeben (remote procedure call). Ein AP ist durch folgende Parameter gekennzeichnet:

$AP(n,j,s,i,o)$

n Name des AP

j Nummer des Knotens, auf dem er implementiert ist

s Sperrtyp = Art der Nutzung und Menge der genutzten Datenmenge

1 - Datensatz lesen

2 - Datensatz schreiben

3 - Datei lesen

4 - Datei schreiben

5 - Datenbasis lesen

6 - Datenbasis schreiben

i Inputparameter

o Outputparameter

s ist in jedem Prozeß (AP) als fixer Parameter gesetzt, kann aber auch, abhängig von der Sperrkategorie der Aktion, im aktivierten Prozeß höher gesetzt werden.

Analoge Informationen (wie über die Prozesse) erhält das VDB über die erforderlichen Daten aus der Datenbasisbeschreibung (DDS), die ebenfalls jedem Knoten bekannt ist.

Über eine Botschaft werden dann die zu einer Aktion A gehörenden Prozesse $P(A)$ systemweit aktiviert, durch die Aktionsnummer a gekennzeichnet und mit Daten versorgt: $P(a,n,j,s,i,o)$.

An dem oft benutzten Beispiel der Umbuchung ist im Anhang der Ablauf einer Aktion unter Steuerung des VDB erläutert.

Prinzip der Netzintegration

Das VDB besteht aus den Komponenten in Abb. 2, die nach dem bekannten Schichtenkonzept strukturiert sind, und durch Übergabe standardisierter VDB-Botschaften miteinander kommunizieren. Diese Botschaften werden bei jedem Übergang durch die entsprechenden Steuerfelder gerahmt (Framing). Die Trennung zwischen typischer Netzwerk- und Datenbankarbeit ist zwischen der 2. und 3. Schicht angesiedelt.

Je nach Art des Prozesses wird der RPC verbindungslos (Prozeß-Nr. 1 bis 29) oder verbindungsorientiert (Prozeß-Nr. ab 30) aufgebaut. Das Kommando *call remote (parameters)*

hat für das VDB folgende einheitliche Parameter-Struktur:

server

Angabe des Zielknotens;

status

ok, absent, unable, not-done;

service

einheitliche Kennung als VDB-Dienst;

length

Länge des funktionsinternen Teils;

time-out

maximale Wartezeit auf die Reaktion; data

eigentlicher VDB-Teil (3-4099 Byte)

- Nummer des Quell-Knotens
- AP-Kennung
- Prozeß-Nummer
- Daten oder Prozeß-Parameter.

Für den RPC ist data nur eine Stringkette, deren interne Struktur im VDB-Kern interpretiert wird (Framing).

Die oben angegebene Schichtenstruktur ist in jedem zum VDB gehörenden Knoten identisch. Dabei ist ein passiver Zustand möglich, in dem ein Knoten nur im Zustand LISTEN (der ersten Schicht d. h. hören am Netz) ist und beim Eintreffen des ersten VDB-Frames, der allerdings durch eine spezielle procedure als solcher erkannt werden muß, die Initialisierung der anderen Schichten veranlaßt (VDB_INIT).

Besonders wichtig für die effektive Funktion des Gesamtsystems ist der Prozeß-Verwalter (Abb. 4). Er wird in folgenden Leistungsstufen implementiert (siehe auch Tab. 1):

1. Er verwaltet nur die auf dem eigenen Knoten existierenden AP und teilt diese nur dem Netz zu. Das setzt bereits voraus, daß diese AP dem Netz bekannt sein müssen. Die AP können nur auf eigene Daten wirken (Datei-Server).

2. Er verwaltet nur die auf dem eigenen Knoten existierenden AP und teilt diese den eigenen Aktionen oder dem Netz zu. Die AP können nur auf eigene Daten wirken. Das setzt die Organisation eines gewissen Hintergrundbetriebes voraus, der dadurch gesichert wird, daß die AP nur über Datenbereiche kommunizieren. Ein- und Ausgaben werden über den speziellen Datentyp *Maske* realisiert.

3. Er verwaltet sowohl die auf dem eigenen Knoten existierenden AP als auch die ihm im Netz bekannten AP und teilt letztere den eigenen Aktionen zu. Die AP können jeweils auf jene Daten wirken, die sich auf demselben Knoten befinden.

4. Er verwaltet sowohl die auf dem eigenen Knoten existierenden AP als auch die ihm im Netz bekannten AP und teilt diese den eigenen Aktionen oder dem Netz zu. Die AP können sowohl auf

Tab. 1 Leistungsstufen des Prozeß-Verwalters

mögliche Zustände	lokaler Knoten		Knoten X		Knoten Y	
	AP	Daten	AP	Daten	AP	Daten
1	x	x				
2			x	x		
3	x			x		
4		x	x			
5			x			x

x = AP oder Daten vorhanden

jene Daten wirken, die sich auf demselben Knoten befinden, als auch auf Daten, die sich auf einem dritten Knoten befinden.

5. Optimierung des Gesamtsystems nach minimaler Belastung der Knoten, minimalem Netzzugriff bzw. minimalem Datentransfer.

Abstrakte Prozesse (AP), die das VDB bereitstellt

Übersicht siehe Tab. 2.

Beispiel: Umbuchung

Zur Aktion *Umbuchung* gehören folgende abstrakte Prozesse (AP):

- AP1 = Lesen Datensatz, s = 1;
- AP2 = Schreiben Datensatz, s = 2;
- AP3 = Eröffnen von Datei mit schreibendem Zugriff auf Satzebene, s = 2; oder
- AP3 = Eröffnen von Datei mit schreibendem Zugriff auf Dateiebene, s = 4;

AP4 = Schließen von Datei mit Rücksetzen von s in DDS.

Daraus und aus der Beschreibung der Datenbasis (DDS) ergibt sich folgender Ablauf:

- P31 = Datei i auf Knoten k eröffnen, in DDS(i,k) s=4 setzen
- P32 = Datei j auf Knoten l eröffnen, in DDS(j,l) s=4 setzen
- P11 = Lesen von Satz m in Datei i auf Knoten k und Setzen von s auf Satzebene durch Eintrag der Aktions-Nummer
- P21 = Schreiben neuen Satz m in Datei i auf Knoten k und Rücksetzen von s im Satz
- P12 = Lesen von Satz n in Datei j auf Knoten l und Setzen von s auf Satzebene durch Eintragung der Aktions-Nummer
- P21 = Schreiben neuen Satz n in Datei j auf Knoten l und Rücksetzen von s im Satz

Tab. 2 Abstrakte Prozesse (AP), die das VDB bereitstellt

n	s	i	o	Funktion
Name des AP	Sperrtyp	Inputparameter	Outputparameter	
DB_OPEN /1	5	User_ID DB_Name	OK	Eröffnen des Zugriffs zu einer Datenbasis
DB_CLOSE /2	0	User_ID DB_Name	OK	Schießen einer Datenbasis
DT_OPEN /3	1	User_ID DT_Name	OK	Eröffnen des Zugriffs zu einer Datei
DT_CLOSE /4	0	User_ID DT_Name	OK	Schließen einer Datei

Verteilte Verarbeitung – Prinzipien und Modelle

DSS_COPY /5	0	User_ID DB_Name	DDS-Dateien	Copieren der aktuellen DDS einer DB
P_CALL /6	0	User_ID	Prozeß-Datei	Broadcast-Call nach im Netz verfügbaren AP
SEARCH_REC /7	1	User_ID Feldname Feldinhalt	Zahl der Sätze	Suchen nach Feldinhalt in einer Datei
GET_REC /8	1	User_ID Feldname Feldinhalt	Datensatz	Suchen eines Satzes nach Feldinhalt in einer Datei
PUT_REC /9	2	User_ID Datensatz	OK	Schreiben eines Satzes in eine Datei
SELECT /10	3	User_ID DT_Name2 Selektionsbedingung	Zahl der Datensätze	Logisches Suchen in einer Datei, Ergebnis ist die spezielle Kopie der Datei auf demselben Knoten
COPY /11	3	User_ID DT_Name	Datei	Kopieren einer Datei
SORT /12	4	User_ID Feldname	OK	Sortieren nach Feldinhalt in einer Datei
DT-APPE /13	4	User_ID DT_Name	OK	Zusammenfassen zweier Dateien gleicher Struktur
STATIST /14	4	User_ID DT_Name	OK	Erzeugen zusätzlicher Datensätze mit statistischen Angaben (Summe, Mittelwert, Maximum, Minimum)
DB_INIT /30	6	User_ID DB_Name	OK	¹⁾ Initialisieren einer DB (nur auf dem lokalen Knoten möglich)
DB_EDIT /70	6	User_ID DB_Name	OK	¹⁾ Strukturänderung einer DB (nur auf dem lokalen Knoten möglich)
DT-DELETE /31	6	User_ID DT_Name	OK	¹⁾ Löschen einer Datei = Strukturänderung einer DB
DT-TRANS /32	6	User_ID DT_Name DB_Name	OK	¹⁾ Transfer einer Datei in eine andere DB; Strukturänderung einer DB
EDIT /50	2	User_ID Feldname Feldinhalt	Datensatz	²⁾ Suchen und Editieren eines Satzes nach Feldinhalt einer Datei
DELETE /51	2	User_ID Feldname Feldinhalt	Datensatz	²⁾ Suchen und Löschen eines Satzes nach Feldinhalt einer Datei
APPEND /52	2	User_ID	Datensatz	²⁾ Suchen und Löschen eines Satzes nach Feldinhalt in einer Datei
VDB_INIT /71	0	User_ID	OK	²⁾ Initialisieren des VDB-Kernes (nur auf dem lokalen Knoten möglich)

¹⁾ Funktionen für den Eigentümer von DB und DT – Datenbank-Administrator

²⁾ Komplexe Funktionen vom Charakter einer Aktion, vom VDB aber als Standard vorgegeben.

P41 = Schließen von Datei i mit Rücksetzen von s in DDS auf k

P42 = Schließen von Datei j mit Rücksetzen von s in DDS auf l.

In diesem Fall hat die Aktion beide Dateien gesperrt, was eine einfache Aktionsbearbeitung im Ausgangsknoten bei Beachtung der Konsistenzbedingungen ermöglicht, jedoch keinen gleichzeitigen Zugriff anderer Aktionen ermöglicht. Die Dateien bleiben auf den beiden betreffenden Knoten auch dann noch gesperrt, wenn die Kontobearbeitung in zeitraubendem Bildschirmdialog erfolgen würde. Das bedeutet: die Abfolge der Prozesse ist mit dem notwendigen Sperrgrad in Übereinstimmung zu bringen. Folgender Ablauf ist deshalb auch möglich:

P31 = Datei i auf Knoten k eröffnen, in DDS(i,k) s=2 setzen

*P11 = Lesen von Satz m in Datei i auf Knoten k und Setzen von s auf Satzebene durch Eintrag der Aktions-Nummer

P21 = Schreiben neuen Satz m in Datei i auf Knoten k und Rücksetzen von s im Satz

P41 = Schließen von Datei i mit Rücksetzen von s in DDS auf k

P32 = Datei j auf Knoten l eröffnen, in DDS(j,l) s=2 setzen

P12 = Lesen von Satz n in Datei j auf Knoten l und Setzen von s auf Satzebene durch Eintrag der Aktions-Nummer

P21 = Schreiben neuen Satz n in Datei j auf Knoten l und Rücksetzen von s im Satz

P42 = Schließen von Datei j mit Rücksetzen von s in DDS auf l.

Falls zum Zeitpunkt des Ablaufes der hier aus zwei Aktionen zusammengesetzten Funktion *Umbuchung* ein anderer Zugriff auf die Dateien i oder j erfolgen soll, sind die Möglichkeiten der Prozeßparallelisierung bei gleichzeitiger Beachtung der leicht veränderten, aber mit der Gesamtfunktion eventuell noch verträglichen Konsistenzbedingungen weit vielfältiger:

1. Eine von beiden Dateien ist immer frei für beliebigen Zugriff.

2. Auf die jeweils genutzte Datei ist der Zugriff einer anderen Aktion möglich,

Bürokommunikation mit MicroNET-80

Uwe Schulze, Axel Wüstemann, Mathias Ohlerich
WPU Rostock, Institut für sozialistische Wirtschaftsführung

wenn s (der konkurrierenden Aktion) = 1 oder 2 ist, und der Zugriff nicht gerade auf denselben Datensatz erfolgen soll. Im letzten Fall muß die konkurrierende Aktion bis zur Freigabe der Datei blockiert werden.

Falls aufgrund der gegebenen Konsistenzbedingung ein zwischenzeitlicher Zugriff auf den noch nicht bearbeiteten zweiten Datensatz verhindert werden soll, ist auch folgende Prozeßfolge möglich:

- P31 = Datei i auf Knoten k eröffnen, in DDS(i, k) $s=2$ setzen
- P32 = Datei j auf Knoten l eröffnen, in DDS(j, l) $s=2$ setzen
- P11 = Lesen von Satz m in Datei i auf Knoten k und Setzen von s auf Satzebene durch Eintrag der Aktions-Nummer
- P12 = Lesen von Satz n in Datei j auf Knoten l und Setzen von s auf Satzebene durch Eintrag der Aktions-Nummer
- P21 = Schreiben neuen Satz n in Datei j auf Knoten l und Rücksetzen von s im Satz
- P42 = Schließen von Datei j mit Rücksetzen von s in DDS auf l
- P21 = Schreiben neuen Satz m in Datei i auf Knoten k und Rücksetzen von s im Satz
- P41 = Schließen von Datei i mit Rücksetzen von s in DSS auf k .

Hierbei ist dann allerdings innerhalb der Applikation eine Zwischenspeicherung des ersten Satzes notwendig.

Bei dem hier unterlegten Modell ist es ebenfalls möglich, von einem nicht sehr leistungsfähigen Knoten eine ganze Aktion oder einzelne Prozesse auf anderen Knoten zu aktivieren, wenn diese nur dort möglich sind (Server), oder dort effektiver bearbeitet werden können. Z. B. ist es sicher günstig, Recherchen bzw. Selektionen den Knoten zu übergeben, auf denen sich die Dateien befinden. Diese Optimierungsaufgabe übernimmt zum Teil das VDB.

Literatur

/1/ Jürgen Nehmer: Softwaretechnik für verteilte Systeme. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo 1985

Die Entwicklung auf dem Gebiet der lokalen Rechnernetze für Arbeitsplatz-computer (PC-Netze) ist in den letzten Jahren stark vom Einsatz sogenannter Netzbetriebssysteme gekennzeichnet.

Netzbetriebssysteme (NOS: Network Operating Systems) sind Erweiterungen lokaler Betriebssysteme, die eine transparente Nutzung entfernt erbrachter Dienstleistungen in einem Netzwerk /1/ zulassen. Transparente Nutzung bedeutet, daß diese Dienste in der gleichen Weise genutzt werden, wie die entsprechenden lokalen. Diese Transparenz wird i. allg. dadurch erreicht, daß dem Bezeichner des entfernten Dienstes ein lokal gültiger Bezeichner zugeordnet wird. Auf diese Weise erscheint ein entfernter Dienst wie ein lokaler.

Entfernt erbracht werden zum einen betriebssystem-spezifische Dienste, wie Datei-, Verzeichnis- und E/A-Geräte-Dienste sowie zum anderen netzwerk-spezifische Dienste, wie Kommunikations- und Verwaltungs-Dienste.

MicroNET-80 ist eine erste Stufe zur Realisierung eines größeren Gesamtkonzeptes. Es lehnt sich in seiner Architektur relativ eng an das NOS CP/Net /2/ von Digital Research an.

Implementierung von MicroNET-80

Entsprechend seiner Rolle im Netz kann ein Arbeitsplatz-Computer entweder eine Arbeitsstation oder ein Server

sein. Für die Erweiterung eines CP/M-kompatiblen Betriebssystems zu einer Arbeitsstation im NOS MicroNET-80 wurden folgende Komponenten implementiert:

- MNIOS (MicroNET Network Input Output System)

Es enthält die Netzzugangsschnittstelle und die Anpassung an die konkrete Netzhardware (in der vorliegenden Implementierung SCOM-LAN).

- MNDOS (MicroNET Network Disk Operating System)

Es realisiert die um Kommunikations-Funktionen und einen Redirektor erweiterte BDOS-Schnittstelle.

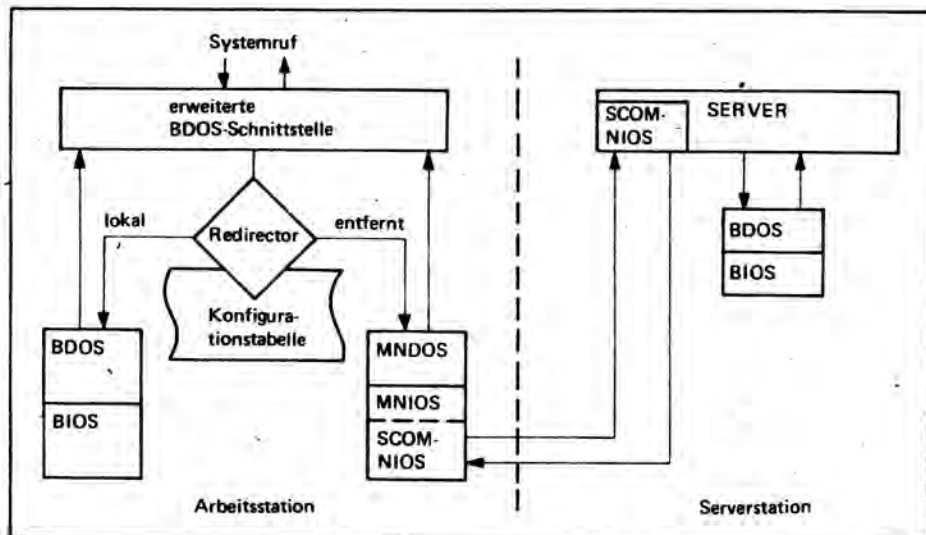
- MNET

Benutzeroberfläche der Arbeitsstation, wahlweise menü- oder kommandogesteuert.

Der *Redirektor* ist ein Modul mit folgenden Aufgaben:

- Zuordnung von Ressourcen eines entfernten Rechners (Servers) zu einem lokalen Gerätebezeichner,
- die Aufhebung dieser Zuordnung, wenn sie nicht mehr benötigt wird
- die Verwaltung der Information über diese Zuordnungen in einer Konfigurationstabelle
- die Umleitung von Systemrufen an entfernte Ressourcen.

Abb. 1 Architektur von Micro-NET-80 unter CP/M-80 auf der Basis von SCOM-LAN



Nr.	Bezeichnung	Input	Output	Bemerkungen
64	Login	Msg-Addr	ReturnCode	Anmelden beim Server
65	Logoff	ServerID	..	Abmelden beim Server
66	Send Message	Msg-Addr	..	Senden einer Nachricht
67	Rcv Message	Msg-Addr	..	Empfangen einer Nachricht
68	GetNwrkStat		Status	Holen Status Byte
69	GetConfAddr		TabAddr	Holen der Adresse der Konfigurations-Tabelle
70	RedirDev	ParamBlock	ReturnCode	Umleiten eines Gerätes zur Netz-Nutzung
71	CancelRed	Device	..	Beendigung der Netz-Nutzung
72	SetIdent	Identifizier	..	Setzen Netz-Identifikation
73	GetIdent		Identifizier	Holen Netz-Identifikation

Funktion	Input	Output	Bemerkung
jp NTRWKINIT		RetCode	Initialisierung des Netzes
jp NTRWKSTATUS		Status	Holen Statusbyte
jp CONFBLADDR		Adresse	Holen Adresse der Konfigurations-Tabelle
jp SNDMSG	MsgAdr	RetCode	Senden Nachricht
jp RCVMSG	MsgAdr	RetCode	Empfangen Nachricht
jp NTRWKERROR			Reinitialisierung nach Fehler
jp NTRWKBOOT			Netz-Warmstart
jp NTRWKID	Identifizier		Setzen Netz-Identifikator

Tab. 1 Erweiterter BDOS-Funktionssatz
Tab. 2 Netzzugangsschnittstelle MNIOS

Der Server ist ein in Turbo-Pascal geschriebenes Anwenderprogramm. Abb. 1 zeigt das Zusammenwirken dieser Komponenten:

Ein von einem Anwenderprogramm in der Arbeitsstation generierter Systemruf wird im Redirektor-Modul des MNDOS daraufhin untersucht, ob er sich auf eine entfernte Ressource bezieht. Dazu greift dieses Modul auf eine Konfigurationstabelle zu, in der vermerkt wird, von welcher Station im Netz welche Ressource verwaltet wird (Laufwerke A: ...P: und List-Geräte). Ist die Ressource entfernt, wird über das Kommunikationssystem eine Anforderung an den Server geschickt, der diese Ressource verwaltet. Nach Erfüllung des Auftrages schickt dieser eine entsprechende Bestätigung zurück. Im Falle, daß die Ressource durch den lokalen Rechner verwaltet wird, wird der Betriebssystemruf wie gewöhnlich im BDOS bearbeitet. Beide Fälle sind

aus Sicht des Nutzerprogramms identisch (transparente Dienstnutzung).

Tab. 1 gibt einen Überblick über den erweiterten Funktionsumfang des BDOS. Die Komponente MNIOS realisiert die Netzzugangsschnittstelle. In Tab. 2 ist diese Schnittstelle angegeben. Sie hat vor allem die Aufgabe, die Gegebenheiten der konkreten LAN-Hardware zu verdecken und so eine Hardware-Unabhängigkeit des NOS zu sichern. Die vorliegende Implementierung nutzt das SCOM-LAN der Ingenieurhochschule für Seefahrt Warnemünde/Wustrow /3/ als Transportsystem. Weitere Generierungen für andere LAN, wie etwa LOTUNET oder ROLANET sind möglich. Sowohl das MNDOS als auch das MNIOS sind in U880-Assembler geschrieben. Sie werden, mit dem MNIOS beginnend, unter das BDOS geladen und beanspruchen 8 KByte des TPA, wobei der größte Teil vom MNIOS beansprucht wird, dessen Größe vom konkreten LAN abhängig ist. Günstiger ist der Einsatz von intelligenten LAN-Adaptern.

Die Benutzerschnittstelle MNET kann

sowohl menü- als auch kommandogesteuert benutzt werden. Abb. 2 zeigt das Grundmenü, Tab. 3 vermittelt einen Überblick über die nutzbaren Kommandos. MNET ist in Turbo-Pascal geschrieben.

Im NOS CP/Net wird als Masterstation das Multi-User-Betriebssystem MP/M benutzt. Daß diese Lösung zumindest für 8-Bit-Technik nicht günstig ist, zeigt die geringe internationale Akzeptanz, die vor allem auf die aufwendige Generierung und Wartung, sowie den geringen TPA (etwa 16 KByte) zurückzuführen ist. Außerdem haben Lösungen, wie Bankumschaltung zur größeren Speicherverwaltung, heute ihre Bedeutung verloren.

Deshalb wurde der transiente Fileserver realisiert.

Der Server fragt abwechselnd die Arbeitsstationen und die Konsole ab. Folgende Dialogfunktionen sind am Server nutzbar:

- Laufwerk selektieren
- Directory anzeigen
- angemeldete Slaves anzeigen
- autorisierte Slaves anzeigen
- Server-Identifikation ändern
- Programm beenden.

Die Arbeitsstationen können sich beim Server ohne Paßwort anmelden (nur Leaseoperationen zugelassen), oder es erfolgt eine autorisierte Anmeldung mit Paßwort, die auch Schreiboperationen gestattet. Der Server wird mit einer Standard-Adresse (Identifikation) generiert; zum gleichzeitigen Betrieb mehrerer Server im Netz kann diese Identifikation geändert werden.

Abb. 2 Grundmenü bei einer Arbeitsstation

MicroNET-80 (c) ISW Rostock Grundmenue
L – An- und Abmelden der Arbeitsstation beim Server N – Definition, Zuruecksetzen und Anzeige von Netzwerk-Ressourcen A – Teilnehmer-Adresse setzen P – Druckdienst K – Uebergang zur Kommandosteuerung
Bitte waehlen sie:

MNET	Laden MicroNET und Menüsteuerung, Aufruf der Menüs
MNET START	Laden MicroNET und Kommandosteuerung
MNET USE	Anzeige der aktuellen Netzkonfiguration
MNET USE äLocDevü=üüRemDevüÄServerIDü	Umleitung eines lokalen Gerätebezeichners auf einen Bezeichner beim Server mit der Identifikation ServerID
MNET USE /R äLocDevü	Zurücknehmen der Benutzbarkeit des Gerätes LocDev im Netzwerk
MNET LOG /I äServerIDü äPasswortü	Anmelden der Arbeitsstation beim Server
MNET LOG /O äServerIDü	Abmelden der Arbeitsstation beim Server
MNET ADDR äIDü	Setzen der eigenen Identifikation auf ID
MNET PRINT äfileü	Drucken der Datei 'file'

Tab. 3 Kommandos von MicroNET-80

Beim Beenden der Arbeit ist zu sichern, daß alle Arbeitsstationen abgemeldet sind, da sonst undefinierte Zustände entstehen. Über ein weiteres Paßwort ist auch dann ein Verlassen möglich.

Erfolgt die Anforderung durch eine Arbeitsstation, so wird die entsprechende BDOS-Funktion ausgeführt und ein Antwortpaket zurückgesandt.

Da die Selektion der Arbeitsstation durch die Transportschicht vorgenommen wird und für die logisch höheren Schichten zufällig erfolgt, werden für alle Arbeitsstationen eigene File Control Blocks (FCB) verwaltet.

Den Anwenderprogrammen werden nur die Restriktionen in bezug auf die Benutzung der BDOS-Funktionen auferlegt, wie sie eigentlich im BDOS existieren (Benutzung der Funktion *Search for Next* (12H) nur nach *Search for First* (14H), weil kein FCB mehr übergeben wird).

In der Tat stellt die Synchronisation dieser beiden Funktionen von verschiedenen Stationen ein Problem dar, da die Nicht-Wiedereintrittsfähigkeit des BDOS umgangen werden muß. Dazu müssen die jeweils zuletzt ausgeführten Funktionen der Arbeitsstation gespeichert werden.

Das Serverprogramm verwaltet alle dem Netz zur Verfügung stehende Peripherie und organisiert den Zugriff durch die Arbeitsstationen. Leseoperationen sind von allen Stationen aus ohne Einschränkungen auf gleiche Diskette und Datei möglich. Bei Schreiboperationen wird

ein File für andere Stationen gesperrt. Eine block- oder satzweise Verwaltung erscheint auf 8-Bit-Technik nicht sinnvoll.

Erfolgt durch das Betriebssystem eine Unterstützung des I/O-Bytes, so ist es möglich, von verschiedenen Stationen gleichzeitig verschiedene Drucker zu nutzen. Dazu muß beim Anmelden der entsprechende Drucker mit seiner Nummer (0 ... 3) angewählt werden. Der Server teilt die Druckvorgänge, was sich auch mit einem Paralleldrucker bewährt hat. Zur Einrichtung einer Druckstation ohne Laufwerke und Konsole steht ein Print-Server als ROM-Version zur Verfügung. Ähnliches wäre auch für eine Festplattenversion denkbar. Das Menü des Servers zeigt Abb. 3.

Ergebnisse

Micro-NET bietet die Möglichkeit, CP/M-Standardsoftware in Netzumgebung abzuarbeiten. In CP/M sind (im Gegensatz zu MS-DOS ab Version 3.1. mit dem SHARE-Befehl) keine Mechanismen zum Sperren und Freigeben (Lock/Unlock) von Files oder Records enthal-

ten. Diese müssen aber auf Betriebssystemebene zur Verfügung stehen, um verteilte Applikationen zu implementieren. Deshalb gilt für Micro-NET die Einschränkung des nicht-nebenläufigen Zugriffs bei Schreiboperationen.

Die Nutzung etwa von dBase im Netz ist zwar möglich, zeigt aber deutlich die Grenzen der Übertragungsgeschwindigkeit auf (da die Realisierung der Schnittstelle auf Betriebssystemniveau eine wesentlich höhere Netzbelastung als eine verteilte Applikation mit sich bringt) und kann die Datenkonsistenz bei nebenläufigen Schreibzugriffen nicht sichern.

Als günstig erwiesen hat sich der Betrieb eines Druckerservers, von dem aus drei Drucker (darunter ein Paralleldrucker) über verschiedene Schnittstellen betrieben werden. Des weiteren ist die Organisation der Druck-Jobs in einer Warteschlange denkbar, sofern das erforderlich ist.

Nützlich ist ein Netzbetriebssystem wie Micro-NET insbesondere für die Ausbildung bzw. den Betrieb von OEM-Hardware ohne Laufwerke, wo vor allem Kopierarbeiten abzuwickeln sind. Die bekannten Dienstprogramme (POWER, NSWEEP) arbeiten im Netz nur unbedeutend langsamer als speziell Netzsoftware für diese Aufgaben, deren Handhabung insbesondere für Gelegenheitsnutzer ungewohnt ist.

Literatur

- /1/ Wüstemann, A.: Netzbetriebssysteme – Eine Technologie zum Verbund von Arbeitsplatzrechnern. In dieser Ausgabe
- /2/ Digital Research: CP/NET User's Guide. Pacific Grove, 1980 (Firmenschrift)
- /3/ Goede, B.; u. a.: Handbuch-SCOM-LAN. Rostock-Warnemünde, Sömmerda 1987

Abb. 3 Menü des Servers

MicroNET-80		
Server-	on CP/M-80	
(C) 1988		

Logged drive: B		
Quit/eXit	Slaves	Dir
Autorised	Identification	

Verteilte Verarbeitung – Datenkonvertierung bei relationalen Datenbanken

Heidrun Ortleb, Wilhelm-Pieck-Universität Rostock
Institut für sozialistische Wirtschaftsführung, Informatik-Labor

Mit der wachsenden Verbreitung von Mikrocomputern in Form von Büro-, Personal- und Heimcomputern bekommen Probleme, die mit Herstellung, Anwendung und Portabilität von Programmen und Daten zu tun haben, eine neue Bedeutung. So kann die Notwendigkeit bestehen, zuverlässig laufende Projekte von einer Rechnerklasse in eine andere umzusetzen oder Daten aus bestehenden Programmsystemen in andere zu übertragen. Massendaten werden oft mittels Datenbanksystem zentral verwaltet.

Es existiert bereits eine Vielzahl von Datenbanksystemen, deren Entwicklung bei weitem noch nicht abgeschlossen ist. Qualitativ neue Probleme stellen höhere Forderungen an die Datenverwaltung, so daß ein bestehendes Datenbanksystem gegebenenfalls erweitert werden muß oder auf ein anderes, für das spezielle Problem besser geeignetes System zurückgegriffen wird. Somit ist es notwendig, aktuelle Daten des einen Systems zu Daten eines zweiten Systems zu transformieren.

Denkt man an die Erfassung von Daten an einem billigen PC und Weiterverarbeitung an einer hochwertigen Anlage, z. B. innerhalb eines Netzes, so kann selbst diese Arbeit den Transport von einem Datenbanksystem zu einem weiteren veranlassen. Den Nutzer eines verteilten System interessiert es primär nicht, ob er über ein Kommunikationssystem, das seinem Datenbanksystem aufgesetzt ist und die verteilte Verarbeitung verwaltet, seine Dateien bearbeitet oder ob er innerhalb seines Datenbanksystems ein Kommunikationssystem benutzt, das ihm den Zugriff zu entfernt liegenden „Fremddateien“ gewährt.

Seine Benutzeroberfläche, also die Benutzersprache, ist brauchbar, wenn sie entweder wenig permanentes Vorwissen verlangt, wie es bei maskenorientierten Sprachen der Fall ist, oder wenn der Benutzer in der ihm gewohnten Programmierumgebung agieren kann, um seine Fähigkeiten effektiv einzusetzen. Somit ist es sinnvoll, die Übertragung mit sprachlichen Mitteln des Datenbanksystems vorzunehmen, unabhängig vom Grad seiner Verteilung.

Datendefinitions- und -manipulationsprachen stellen mit ihren Statements weitaus komplexere Anweisungen zur Definition von Datentypen und Operation mit ihnen zur Verfügung als die alten Sprachen wie z. B. FORTRAN und COBOL, wo sich nur der einzelne Record des Files ansprechen läßt. Sie können mit einem Befehl das ganze File aufrufen, und die knifflige Sonderprogrammierung für den ersten und letzten Satz entfällt. Oft können auch Zwischenergebnisse bei Gruppenwechsel durch Variable, deren Werte die Gruppenzugehörigkeit kennzeichnet, gespeichert werden.

Die Art der Transformation hängt selbstverständlich von den logischen Datenmodellen der Quell- und der Zieldatenbank ab. Sind beide gleicher Art, so wird die Struktur der Daten logisch übernommen, ansonsten muß auch sie selbst transformiert werden. Prinzipiell lassen sich hier unterschiedliche Niveaustufen ableiten. Um Daten einer Rechnerklasse in einer anderen verarbeiten zu können, ist i. allg. eine Konvertierung notwendig, so daß ein Zeichen (eine Ziffer) als dasselbe (dieselbe) am anderen Computer erkannt wird. Diese Art Konvertierung soll *Konvertierung nullter Stufe* heißen. Sie ist Voraussetzung jeder weiteren Konvertierung.

Eine Konvertierung erster Stufe besteht dann, wenn Formatangaben und Satzbeschreibungen (die nötigen Daten des Data-Dictionary) gleichzeitig mit konvertiert werden. Damit entfallen beim Lesen und Schreiben alle Formatanweisungen, Angabe des Files oder der Variablen genügen.

Werden nun Strukturen in andere Strukturen konvertiert, dann entspricht das einer Konvertierung zweiter Stufe. Der einfachste Fall liegt vor, wenn Daten einer relationalen Datenbank in Daten einer zweiten, wiederum relationalen Datenbank, umgeformt werden. Hier genügt eine Konvertierung erster Stufe. Auf diesem Niveau bieten sich zwei Möglichkeiten zur Übertragung von Datenbankdateien an: Der Benutzer wird über Masken eindeutig geführt und kann prinzipiell nichts falsch machen.

Andererseits kann er in der Datenmanipulationssprache seines Systems, die möglicherweise selbst als Maskensprache realisiert ist, Zugang zu Daten des Fremdsystems erhalten.

Soll in ein Datenbanksystem A eine Teildatenmenge des Systems B übernommen werden, so kann die Beschreibung der gesuchten Menge in der Sprache des Systems A erfolgen. Die dazu nötigen Statements werden in die Datenmanipulationssprache des Systems B übersetzt und hier ausgeführt. Das Ergebnis der Anfrage, also Daten mit Metadaten, wird in das Zieldatenbanksystem konvertiert. Damit benötigt der Nutzer nur Kenntnisse über das Datenbanksystem, mit dem er arbeitet und es wird in den beteiligten Datenbanksystemen mit systemeigenen Mitteln gearbeitet.

Praktisch wurden am ISW Konvertierung von TOPAS in DES, TOPAS in dBASE, dBASE in DES vorgenommen. Konvertierungen von dBASE in DES oder TOPAS in DES können dabei über ein Menü im Initialisierungsteil DBINIT des Datenbanksystems oder mittels Sprachanweisung erfolgen.

Für DES wird eine Minisprache entwickelt, deren SELECT-Klausel zur Formulierung von Anfragen durch ein Syntaxdiagramm in PASCAL-Notation beschrieben ist (Abb. 1).

Wie in der PASCAL-Syntax stehen Terminale in Kästchen und Nichtterminale in Ovalen. Dabei bedeuten: relneu, relalt – Variablenamen für Datenbankdateien; feld – ein beliebiger Feldname, literal steht für eine Zeichenkette und num sei eine vorzeichenlose Zahl.

Zum Beispiel soll eine Anfrage an eine TOPAS-ESER-Datei gestellt werden. Hier erfolgen lexikalische und syntaktische Analyse durch den gleichen Modul wie im Fall einer Anfrage an DES, nur hier wird im Ergebnis der Synthese anstelle des Moduls zum Finden der gesuchten Datensätze eine neue Anfrage in Form von TOPQUERY erzeugt. Erst die so erzeugte TOPAS-Anfrage gelangt mittels Fileserver im Fall einer online-Kopplung in die ESER-Anlage. Das Ergebnis beinhaltet zwei Dateien, eine Beschreibungsdatei, in der die Datenfor-

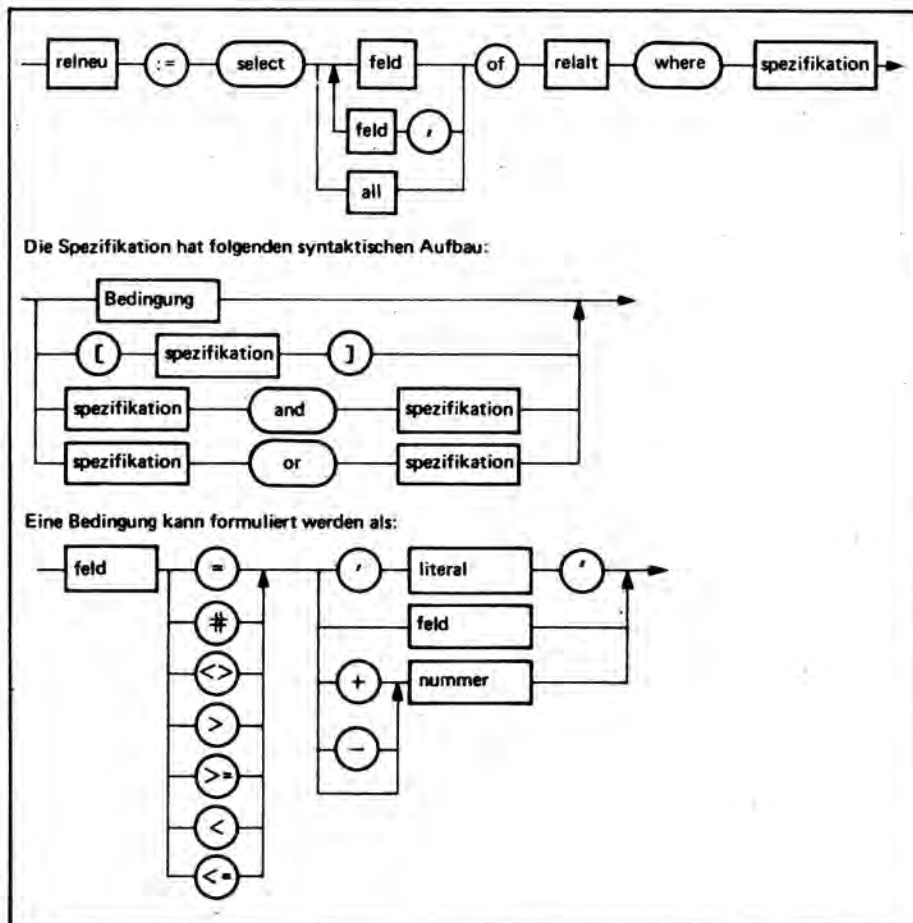


Abb. 1 Syntaxdiagramm zur Formulierung von Anfragen (PASCAL-Notation)

mate der TOPAS-Datei in PL/1-Form beschrieben sind und eine Liste, in der die Werte der einzelnen Felder (Spalten) enthalten sind. Beides wird wieder im PC bearbeitet. Aus der ersten Datei wird die Struktur der entstehenden DES-Datei entwickelt, während die zweite zum Füllen der DES-Datei dient. Selbstverständlich werden dazu die Textdaten entsprechend ihrem Typ konvertiert.

So wird z. B. eine DES-Anfrage der Form: `dat2 := select feld1, feld2, feld5 of dat1 where (feld1 < 10 or feld4 = 'Wert') and feld3 # 4.5;` umgeformt in folgende Anfrage:

FOR DAT1

```
SELECT DAT1.FELD1, FELD2, FELD5
WHERE (DAT1.FELD1 < 10 !
DAT1.FELD4 = 'WERT') & FELD3 <> 4.5
PLIDES
LIST.
```

Sofern die Schnittstelle TQL für TOPAS fertiggestellt ist, d. h. sofern mit TOPAS-Daten in einer dBASE-ähnlichen Form *gesprochen* werden kann, wird dasselbe Ergebnis erreicht, wenn der Modul zur Erzeugung von TOP-QUERY-Anweisungen durch den Modul zur Erzeugung von dBASE-Anweisungen ausgetauscht wird. In diesem Fall können auch die Ergebnisdaten in dBASE-Form direkt an den Ziel-PC übertragen und die entsprechende Wandlung dBASE-DES genutzt werden.

aspekte-vorschau 3/89

REDABAS-Beispiele für 8- und 16-Bit-Rechner

Um den zahlreichen Nutzern von REDABAS eine Anleitung zur Nutzung des weit verbreiteten Softwareangebotes zu geben, ist diesem Thema die *edv-aspekte* 3/89 gewidmet. Dabei richtet sich das Heft sowohl an Leser, die mit REDABAS noch wenig Erfahrungen haben als auch an den Nutzerkreis, der seine Kenntnisse zu dieser Thematik vertiefen möchte.

Aus dem Inhalt:

- Beispiele zum Eröffnen von REDABAS-Programmsystemen (u. a. auch Inversdarstellung, Blinken, usw., Zeitschleifen, Nutzungscode abfragen usw.)
- Menügestaltung und Menüauswertung
- Dateiarbeit (Dateinamen erfragen, Laufwerk erfragen, Dateiattribute setzen und abfragen)
- Dateistrukturbeschreibungsdateien und ihr Einsatz
- Datenerfassung und Datenpflege
- Blättern in REDABAS-Datenbanken (bei Textdateien, bei allgemeinen Dateien)
- Programmieren in REDABAS für REDABAS (Bedingungsgenerator, Feldlistenentwicklung, Makroersetzungen, etc.)
- Programmieren in REDABAS für andere Systeme (Kalkulationsprogramm, Textprogramm, PC-GRAPH, BASIC)
- Kopplung von REDABAS mit anderen Systemen (Entwicklung integrierter Systeme)
- (Senden und Empfangen von Daten zu bzw. von anderen Systemen, offline über ASCII-Dateien, online über QUIT TO ..., Einsatz von SUBMIT-Aufgabendateien, Schleifen bei SUBMIT-Aufgabendateien, Übernahme von Daten aus anderen Systemen, Übergabe von Kommandodateien für andere Systeme)
- Windowtechniken und Benutzung von Farben
- Lokales Netz und verteilte Ressourcen.

Rechnergestützte Arbeit mit Mitteilungen im Büro

Frank Försterling, Ingo Kerlikowski, Prof. Dr. Claus Sattler,
Dr. Christine Teßmer, Institut für Informatik und Rechentechnik der AdW

Der wachsende Einsatz von Personalcomputern im Büro hat dazu beigetragen, eine Reihe von Bürotätigkeiten zu rationalisieren. Dabei wurden auf der Basis von Standardsoftware auf die zu lösende Aufgabe zugeschnittene Anwendungssysteme geschaffen. Textverarbeitung, Datenbankarbeit, Grafik sowie Kalkulationssysteme werden genutzt, um Sekretariatsarbeiten und Sachbearbeiterarbeiten zu erleichtern.

Ein Tätigkeitsbereich, der einen hohen Anteil an der Büroarbeit hat, wurde zumeist ausgeklammert: die Kommunikation.

Darüber täuscht auch ein rechnergestütztes Postein-/ausgangsbuch nicht hinweg. Hauspost und Briefverkehr erfolgen traditionell. Die Ursachen dafür liegen in der zumeist noch fehlenden Vernetzung. Mit zunehmender Bereitstellung technischer Mittel zur lokalen und weiträumigen Vernetzung auf der Basis lokaler Netze und Datennetze fallen diese Schranken weg, und es entsteht die potentielle Möglichkeit der Bürokommunikation.

In /1/ werden verschiedene Formen der Bürokommunikation systematisiert und vergleichend betrachtet.

Im weiteren wird ein verteiltes Briefverwaltungssystem als eine spezielle Form eines rechnergestützten Mitteilungensystems vorgestellt, das neben Funktionen zur Kommunikation in Form des Sendens und Empfanges von Briefen weitere Bearbeitungsfunktionen mit einbezieht. Dazu gehören die Brieferstellung sowie die Integration der Arbeit mit der elektronischen Ablage und dem elektronischen Postein-/ausgangsbuch. Bezüglich der bereitgestellten Kommunikationsfunktionen realisiert es eine Ein-Weg-Kommunikation für Textinformationen mit Möglichkeiten für die Individualkommunikation und das Verteilen.

Modell der Arbeitsvorgänge

Betrachten wir die Arbeitsvorgänge bei der Arbeit mit der Post im Büro, zunächst das Anfertigen und Versenden. Ein Brief wird mit dem Textverarbeitungssystem angefertigt und soll anschließend versendet werden. Mit dem

Versenden sind folgende Entscheidungen bzw. Handlungen verbunden:

- Festlegung der Empfänger von Originalen,
- Festlegung der Empfänger von Kopien und der Bemerkungen, mit denen die Kopien zu versehen sind,
- Festlegung der Ablagefächer, in denen eine Kopie des Briefes aufzubewahren ist,
- Eintragen der ausgehenden Briefe in das Postausgangsbuch.

Bei der Festlegung von Empfängern von Originalen und Kopien wird mit Adreßverzeichnissen und Verteilerlisten gearbeitet. Für jeden Brief ist der Kopfbogen auszufüllen und ein Briefumschlag vorzubereiten.

Eventuell ist ein Brief vor dem Absenden noch mit anderen Mitarbeitern (z. B. dem Leiter) abzustimmen.

Bei Posteingang sind (durch die Sekretärin) zunächst folgende Handlungen durchzuführen:

- Eintragen des eingegangenen Briefes in das Posteingangsbuch,
- Einsortieren des Briefes in die Postmappe.

Die Postbearbeitung besteht im wesentlichen aus folgenden Handlungen:

- Auswählen der zu bearbeitenden Post, sofern nicht alles sofort erledigt werden kann oder soll,
- Lesen des Briefes,
- Bearbeiten des Briefes.

Das Bearbeiten des Briefes kann in unterschiedlicher Weise erfolgen und folgende Handlungen beinhalten:

- Weiterleiten des Briefes bzw. von Kopien an andere Mitarbeiter, eventuell mit Bemerkungen, was diese mit dem Brief tun sollen,
- Beantworten des Briefes,

- Einordnen des Briefes in die Wiedervorlagemappe mit einem bestimmten Datum,

- Einordnen des Briefes in Ablagefächer,
- Liegenlassen des Briefes zur späteren Bearbeitung.

Darüber hinaus gibt es eine Vielzahl weiterer Arbeitsvorgänge, von denen hier nur betrachtet werden sollen:

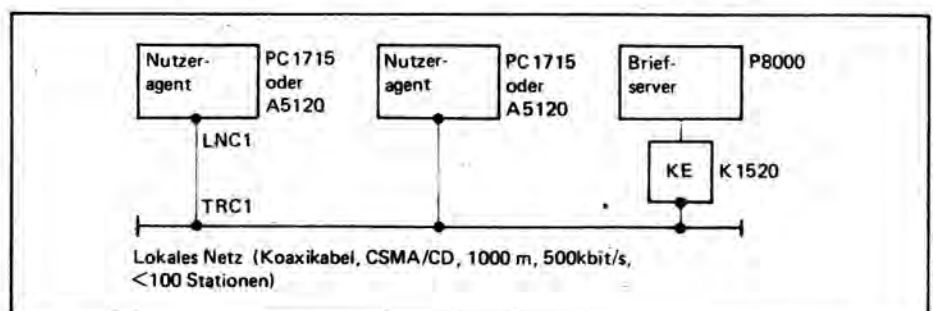
- Anlegen und Ändern von Adreßverzeichnissen und Verteilerlisten,
- Suchen eines Briefes in einem der Ablagefächer,
- Suchen eines Briefes im Postein-/ausgangsbuch,
- Wiedervorlage von Briefen nach Ablauf der Wiedervorlagefrist.

Ausgangssituation und Hauptmerkmale

Aufgabe bei der Gestaltung des verteilten Briefverwaltungssystems war es, die Übertragung der Briefe über Rechnernetze zu ermöglichen und die im Modell der Arbeitsvorgänge aufgeführten Handlungen rechnergestützt durchführen zu können. Bei der Festlegung der Netzstruktur wurde von einer Teilung in interne Post (hier zumeist nicht in Form von Briefen, sondern von Mitteilungen) und Post zwischen Einrichtungen ausgegangen. Interne Kommunikation wird zukünftig zunehmend im Rahmen von lokalen Netzen erfolgen, externe Kommunikation unter Nutzung eines Datennetzes.

Im Büroarbeitsplatzbereich werden in nächster Zeit noch verstärkt 8-Bit-Personalcomputer zum Einsatz kommen,

Abb. 1 Prinzipielle Hardware-Struktur eines verteilten Briefverwaltungssystems



die neben Kommunikationsaufgaben auch für die Bearbeitung lokaler Aufgaben genutzt werden und durch die Spezifik der Betriebssysteme (kein Multitasking) in diesen Zeiträumen nicht für Kommunikationsaufgaben (z. B. den Empfang von Briefen) zur Verfügung stehen.

Daraus ergibt sich als Konsequenz die Nutzung eines Servers, auf dem für eine bestimmte Struktureinheit (Abteilung, Einrichtung) die elektronischen Briefkästen, Ablagefächer und Postein-/ausgangsbücher angesiedelt sind und zu dem bei der Arbeit mit der Post zugegriffen wird. Server verschiedener Bereiche sind dann miteinander zu verbinden, entweder innerhalb des lokalen Netzes oder über ein Datennetz.

Die prinzipielle Hardware-Struktur eines solchen verteilten Briefverwaltungssystems ist in Abb. 1 dargestellt.

Hardware und Kommunikationssoftware

Das realisierte System geht davon aus, daß im Büro Personalcomputer PC 1715 oder Bürocomputer A 5120 genutzt werden. Der Server ist derzeit auf der Basis des P8000 realisiert. Die lokale Verkopplung erfolgt auf der Basis der Hardware von ROLANET1.

An einer vergleichbaren Lösung für 16- und 32-Bit-Technik wird gearbeitet.

Die Anwendungssoftware für das verteilte Briefverwaltungssystem nutzt eine OSI-gerechte Kommunikationssoftware für die Schichten 1 bis 4. Insbesondere sind in der Schicht 3 das X.25-Packet-Level-Protocol /2/ und in der Schicht 4 ein Transportprotokoll der Klasse Null /3/ realisiert. Damit soll aus der Sicht einer Forschungseinrichtung, die sich mit OSI-gerechter Kommunikation befaßt, im lokalen Netz die Voraussetzung für ein OSI-gerechtes Gateway zu einem Datenpaketnetz geschaffen werden /4/.

Die für das verteilte Briefverwaltungssystem genutzte OSI-gerechte Kommunikationssoftware ist simultan mit der Kommunikationssoftware von ROLANET1 auf einem Kabel betreibbar. Um die Einbindung von ROLANET1-Büroarbeitsplätzen weiter zu verbessern,

```

Brief_t      = RECORD
                Briefumschlag : Briefumschlag_t;
                Briefinhalt   : Briefinhalt_t
            END;
Briefinhalt_t = RECORD
                Briefkopf     : Briefkopf_t;
                Briefftext    : Briefftext_t
            END;
Adresse_t    = RECORD
                Struktureinheit : char_6;
                Name           : char_14
            END;
Briefumschlag_t = RECORD
                Absender       : Adresse_t;
                Empfaengerliste : ARRAY 1..n OF Adresse_t;
                Sendedatum     : char_8;
                Zeit           : char_5;
                Prioritaet     : (dringend, eilig, normal);
                Quittung       : BOOLEAN
            END;
Briefkopf_t  = RECORD
                Original        : BOOLEAN;
                Unser_Zeichen   : char_14;
                Ihr_Zeichen     : char_14;
                Unsere_Journal_Nr : integer_5;
                Ihre_Journal_Nr  : integer_5;
                Ihr_Datum       : char_8;
                Ihre_Zeit       : char_5;
                Vermerkliste    : Vermerkliste_t;
                Betreff         : char_50
            END;
Vermerkliste_t = ARRAY 1..m OF Vermerk_t;
                = (Ruecksprache, Kenntnisnahme,
                  Erledigung, Stellungnahme,
                  Entscheidung, Verbleib,
                  Rueckgabe, Bemerkung)
    
```

wird gegenwärtig eine Umstellung der Kommunikationssoftware auf ROLANET1 geprüft.

Der P8000 ist nicht direkt, sondern über eine Kommunikationseinrichtung auf der Basis K 1520 an das lokale Netz angeschlossen. Das ist dadurch begründet,

Abb. 2 Struktur eines Briefes

daß für den P8000 kein ROLANET1-Controller zur Verfügung steht. P8000 und K 1520 sind über eine serielle Schnittstelle gekoppelt.

Brief-Nr.: 127

Empfänger		Absender	
Struktureinheit /	Name	Struktureinheit /	Name
Forschung /	Schulze	Entwicklung /	Mueller
Ihr Journal /	Ihr Zeichen	Unser Journal /	Unser Zeichen
202	Schulze	507	Mueller
: 10.07.88/10:36		Vom: 12.07.88	
Prioritaet: N		Vermerk: BM	
betreff: Thema 1.1.			

Abb. 3 Anzeige von Umschlag und Briefkopf auf dem Bildschirm

Briefstruktur

Ein Brief besteht normalerweise aus dem Umschlag und dem Inhalt, wobei bei geschäftlicher Post der Inhalt einen Kopfbogen enthält. Der Kopfbogen besteht aus dem Briefkopf und dem Brieftext. Sowohl der Umschlag als auch der Briefkopf setzen sich aus mehreren Bestandteilen, den Attributen, zusammen. Attribute des Umschlags sind z. B. Struktureinheit und Name. Attribute des Briefkopfes sind z. B. Datum, Betreff, Journal-Nummer, Unser Zeichen. Bei der Beantwortung von Briefen werden auch solche Attribute benutzt wie Ihre Journal-Nummer, Ihr Brief vom, Ihr Zeichen. Abb. 2 zeigt die in unserem System verwendete Struktur eines Briefes. Abb. 3 enthält die Anzeige der Attribute von Umschlag und Briefkopf auf dem Bildschirm.

Gestaltung des Servers

Auf dem Server sind die elektronischen Briefkästen, Ablagefächer und Postein-/ausgangsbücher für die Nutzer organisiert, die über diesen Server verwaltet werden sollen. Jeder Nutzer ist einer Struktureinheit zugeordnet. Dabei ist nicht festgelegt, was eine Struktureinheit repräsentiert. Das kann eine Abteilung, ein Bereich oder eine ganze Ein-

richtung sein. Auf einem Server werden mehrere Struktureinheiten mit ihren Nutzern verwaltet. Es wird jedoch vorausgesetzt, daß alle Nutzer einer Struktureinheit auf einem Server verwaltet werden.

Für jeden Nutzer wird ein Briefkasten angelegt, in dem die für ihn eingetroffenen Briefe sowie die zur Wiedervorlage vorgesehenen Briefe eingeordnet sind und zu dem er nur persönlich Zugriff hat. Zur Wiedervorlage vorgesehene Briefe werden dem Nutzer nach Ablauf der Wiedervorlagefrist wie die eingetroffenen Briefe beim Lesen im Briefkasten mit angezeigt.

Jeder Nutzer kann sich persönliche Ablagefächer definieren, in die Briefe vor dem Senden oder nach dem Empfangen eingeordnet werden können. Die Ablage in mehrere Fächer ist möglich. Zu seinen Ablagefächern kann nur der Nutzer persönlich zugreifen.

Für jede Struktureinheit existiert ein Postein-/ausgangsbuch, zu dem die Nutzer der Struktureinheit zugreifen können, für die eine Zugriffsberechtigung vorhanden ist. Der Zugriff beschränkt sich beim Postein-/ausgangsbuch auf das Lesen.

Neben diesen nutzer- bzw. struktureinheitsbezogenen Objekten gibt es einen Ausgangsbriefkasten für die Briefe, die noch an Nutzer zu versenden sind, die durch andere Server verwaltet werden. Dabei wird davon ausgegangen, daß

nicht zu jedem Zeitpunkt die Kommunikation zwischen beliebigen Servern verfügbar sein muß. Auch aus ökonomischen Gründen kann es sinnvoll sein, zunächst Briefe für einen anderen Server zu sammeln und dann hintereinander zu übertragen.

Die Verwaltung der beschriebenen Objekte erfolgt über ein speziell für die Speicherung dokumentenorientierter Objekte realisiertes Datenverwaltungssystem. Darauf wurde im Ergebnis einer Analyse orientiert, die ergab, daß bestehende Datenbank- und Datenverwaltungssysteme den Anforderungen der Speicherung und Verwaltung dokumentenorientierter Systeme ungenügend entsprechen /5/. Das realisierte Datenverwaltungssystem enthält dynamische Satzsegmente und einen Transaktionsmechanismus.

Die Speicherung der Objekte erfolgt auf einer Festplatte. Die Sicherung des Inhalts der Festplatte muß durch administrative Maßnahmen außerhalb der Nutzungszeit des Servers erfolgen.

Für die Verwaltung des Servers ist eine umfangreiche Administrator-Software verfügbar, die das Anlegen und Verändern der beschriebenen sowie weiterer Objekte, Zustandsabfragen sowie die Ausgabe von Statistik über die Arbeit des Systems interaktiv im laufenden System ermöglicht. Das Grundmenü für den Administrator ist in Abb. 4 dargestellt. Dabei wird davon ausgegangen, daß eine Person als Administrator für den Server fungiert. Gegenwärtig ist die Bedienung der Administrator-Funktionen eines Servers nur lokal möglich, an Problemen der entfernten Nutzung der Administrator-Funktionen wird gearbeitet.

Der Server ist auf der Basis des Betriebssystems WEGA für den P8000 realisiert. Er umfaßt mehrere Betriebssystem-Prozesse. Die parallele Nutzung des P8000 durch andere Anwendungen mit direkt über das lokale Netz angeschlossenen Terminals ist im Rahmen der Leistungsfähigkeit des P8000 prinzipiell möglich.

```

*                               - Grundmenue 1 -
*   I - Initialisieren BVS      A - Auskunft      H - Hilfe
*   S - Starten BVS            K - Statistik      E - Ende
*   V - Veraendern BVS        P - Passwort aendern

```

Abb. 4 Grundmenü für den Administrator

*	- Grundmenue(Gm) -	
* F-TP	8-Postbuch lesen	P-Passwort aendern
* S-Brief senden	A-Ablage lesen	D-Adressdatei
* L-Brief lesen	E-Ende	H-Hilfe

Abb. 5 Grundmenü für den Nutzer

Gestaltung der Nutzerschnittstelle

Die Nutzerschnittstelle wird auf 8-Bit-Personalcomputern bereitgestellt. Zur Erfüllung der Anforderungen des Nutzers erfolgt durch die Software eine Kommunikation mit dem Server. Nach dem Starten des Systems erfolgt die Anmeldung des Nutzers mit Paßwort beim Server, die sichert, daß kein Fremder unberechtigt Briefe unter dem Namen des Nutzers absenden kann und kein Fremder unberechtigt Zugriff zum Briefkasten, den Ablagefächern und dem Postein-/ausgangsbuch nehmen kann. Jeder Nutzer hat nur zu den Briefen Zugriff, die entweder von ihm verschickt oder an ihn adressiert wurden. Im Ergebnis eines erfolgreichen Anmeldens erscheint eine summarische Anzeige der im Briefkasten enthaltenen Briefe. Der weitere Ablauf der Arbeit des Nutzers ist menügesteuert. Es erscheint das in Abb. 5 dargestellte Grundmenü sowie im Ergebnis von Nutzerhandlungen weitere Menüs. Dabei sind die genannten Arbeitsabläufe ausführbar.

Die Erfahrungen eines ersten Erprobungseinsatzes zeigen, daß der Gestaltung der Nutzerschnittstelle große Aufmerksamkeit zu schenken ist. Die Handhabbarkeit des Systems hängt dabei u.a. ab von der Anordnung der möglichen Aktionen, der Einstellbarkeit von Nutzerprofilen, einer Minimierung der notwendigen Eingaben, der Rückkehr aus den Menüs.

Anwendungsmöglichkeiten

Zunächst ist festzustellen, daß bei einem derzeitigen System ähnlich wie beim Telefon die Nutzung weitgehend davon beeinflußt wird, ob alle Kommunikationspartner über das System erreichbar sind. Da das nicht sofort der Fall sein wird, benötigt man eine Strategie für den Einsatz, die in engem Zusammenhang mit der beabsichtigten Vernetzung steht. In unserem Institut wurde auf der Basis von ROLANETI ein lokales Netz für die Bürokommunikation aufgebaut, über das die Sekretariate der ersten Leitungsebene verbunden sind. Daraus ergeben sich auch die Einsatzmöglichkeiten. Einladungen zur Dienstberatung, Protokolle der Dienstberatung, Protokolle des Planrapports, Terminkontrollisten sind Beispiele von Mitteilungen (Briefen), die über das System übertragen werden können. Dabei haben wir die Erfahrung gemacht, daß solche Einsatzmöglichkeiten geplant und vorgegeben werden müssen. Es ist günstig, wenn zunächst ein Teilnehmer (oder wenige) aktiv anderen Briefe schickt. In unserem Falle war dies das wissenschaftliche Sekretariat des Direktors. Das ermöglicht auch eine Konzentration der Betreuungskapazität, die in jedem Fall bei der Einführung eines Systems erforderlich ist.

Schlußfolgerungen

Das entwickelte Briefverwaltungssystem stellt eine experimentelle Lösung eines Bürokommunikationssystems dar, mit der praktische Erfahrungen zur Konzipierung und Realisierung derartiger Systeme gesammelt wurden und das erste Anwendungsuntersuchungen ermöglicht. Bei der weiteren Ausgestaltung des Systems ist die Verwendung internationaler Standards ohne grundlegende konzeptionelle Änderungen möglich.

Die Anwendung von 8-Bit-Mikrorechnersystemen als Benutzeragenten zeigt, daß der für den implementierten Funktionsumfang erforderliche Programmcode die Grenzen dieser Systeme erreicht und eine Integration weiterer Anwendungs- bzw. Kommunikationsfunktionen nur in geringem Maße möglich ist. Zukünftig ist für Benutzeragenten der Einsatz von 16-Bit-Mikrorechnersy-

stemen oder von Mehrplatzsystemen vorzusehen. Die Kopplung der Benutzeragenten mit dem Brief-Server über ein lokales Netz mit einer Übertragungsgeschwindigkeit von 50 KBit/s erweist sich für die vorliegende Anwendung (Text, kurze Briefe) als ausreichend.

Literatur

- /1/ C. Sattler: Dienste für die Mensch-Mensch-Kommunikation im Büro. rechentechnik/datenverarbeitung, Manuskript in der Redaktion
- /2/ Information Processing Systems - Open Systems Interconnection - X.25 Packet Level Protocol for Data Terminal Equipment, DP8208
- /3/ Information Processing Systems - Open Systems Interconnection - Connection-oriented Transport Protokoll Specification, IS8073
- /4/ W. Blume, U. Hofmann: Verkopplung lokaler Netze mit Weitverkehrsnetzen. rechentechnik/datenverarbeitung, 10/88
- /5/ F. Försterling: Data Modelling Aspects in Message Systems, zu erscheinen in: Proceedings of the 11th International Seminar on Database Management Systems, Seregelyes/Hungary, Oct. 1988

Auftragstransfer – ein Anwendungsdienst im ESER-Rechnerverbund

Dr. Gerald Hartung
VEB Mikroelektronik „Karl Marx“ Erfurt

Den Nutzern von Rechenzentren (RZ) wird für die Lösung von Aufgaben der Informationsverarbeitung an zentralisierten Rechnersystemen (RS) der Zugang zu ihren Ressourcen über die Betriebsarten Stapelverarbeitung und Dialogverarbeitung ermöglicht. Das Ziel der Stapelverarbeitung besteht in der Erhöhung des Systemdurchsatzes, das der Dialogverarbeitung in der Minimierung der Antwortzeit. Eine Möglichkeit, beide Betriebsarten mit hoher Effektivität zu betreiben, besteht in der funktionellen Trennung der Betriebsarten auf autonome RS. Die funktionelle Trennung von Stapel- und Dialogverarbeitung auf autonome RS wirkt dem ständig wachsenden Integrationsgrad in der Nutzung beider Betriebsarten entgegen. Dieser Widerspruch kann durch den Verbund der funktionell spezialisierten, autonomen RS zu einem Rechnerverbundsystem (RVS) gelöst werden. Die verteilte Anwendung Auftragstransfer (AT), die die funktionellen Möglichkeiten der Dialog- und Stapelverarbeitung vereint, wird als eine anwendungsseitig motivierte Gesamtfunktion für ein RVS definiert (dominierend ist die Erfüllung des Motivs *Funktionsverbund*). Die Funktionalität dieser Anwendung ermöglicht es,

Aufträge von einem beliebigen RS an ein Stapelverarbeitungssystem zu transferieren,

- sie an diesem zu bearbeiten,
- ständig Informationen zum Stand der Auftragsbearbeitung bereitzustellen und
- gegebenenfalls die im Ergebnis der Bearbeitung bereitgestellten Resultate einem beliebigen oder dem bearbeitenden Nutzer zurückzuführen.

Dabei ist unter einem Auftrag eine inhaltlich zusammengehörende Menge von selbständig abarbeitbaren Jobs, die jeweils geschlossen an einem Stapelverarbeitungssystem bearbeitet werden, zu verstehen.

Zur Beschreibung und Strukturierung der Kommunikationsbeziehungen verteilter Anwendungen hat sich das Open Systems Interconnection – Basic Reference Model (OSI-RM) /1/ international durchgesetzt. Obwohl für die Belange der globalen Kommunikation ent-

wickelt, ist dieses Modell für die Beschreibung und Strukturierung der verteilten Anwendung AT anwendbar.

Spezifikation der Anwendung Auftragstransfer

Komponenten des Anwendungsdienstes

In Anlehnung an internationale Standardisierungsbestrebungen zur netzweiten Bearbeitung von Aufträgen – Job Transfer and Manipulation (JTM) /2/ – wurde die verteilte Anwendung AT gemäß OSI-RM als Schicht-7-Anwendungsdienst spezifiziert.

Der Anwendungsdienst wurde in die vier Komponenten

- Auftragserzeugung (K1),
- Auftragsmodifikation (K2),
- Auftragsauswahl und -fortschrittskontrolle (K3) und
- Auftragsverwaltung (K4) gegliedert.

Jede der Komponenten erbringt gegenüber den Dienstenutzerprozessen, die in der Problemumgebung lokalisiert sind, durch kommunikative Wechselwirkung mit ihrer Partnerkomponente einen definierten Funktionsumfang.

Die genannten Komponenten realisieren folgenden Funktionsumfang.

K1: Auftragserzeugung

Sie kommuniziert unmittelbar mit einem Endnutzer, der im Dialog über ein Terminal den Zugang zur Anwendung AT aktiviert, um:

- Aufträge an ein Stapelverarbeitungssystem zu übergeben,
- sich den aktuellen Bearbeitungszustand eines Auftrages anzeigen zu lassen, und
- die Rückführung der Verarbeitungsergebnisse abzufordern.

K2: Auftragsmodifikation

Mittels dieser Komponente wird dem Bediener die Möglichkeit gegeben, Aufträge, in deren Bearbeitungsphase Fehler (Hard- oder Software) auftreten, so zu modifizieren, daß sie von einer weiteren Bearbeitung ausgeschlossen bzw. in einen wiederanlauffähigen Wartezustand überführt werden können.

K3: Auftragsauswahl und -fortschrittskontrolle

Diese Komponente wird benötigt, um:

- unter Berücksichtigung verschiedenster Aspekte (Termin, Ressourcen, etc.) die Auswahl von Aufträgen auf dem zugehörigen RS zu initiieren, und

- die in der Bearbeitungsphase eines Auftrages eintretenden Ereignisse, die den Bearbeitungszustand (Status) eines Auftrages einschließlich all seiner Elemente (Jobs) konkret fixieren, zu registrieren.

K4: Auftragsverwaltung

Aufgabe dieser Komponente ist es, Aufträge RVS-weit zu verwalten. Zu dieser Verwaltungsfunktion zählt:

- die Registrierung von Aufträgen einschließlich der zugehörigen Jobs, die dem Anwendungsdienst AT übergeben werden.

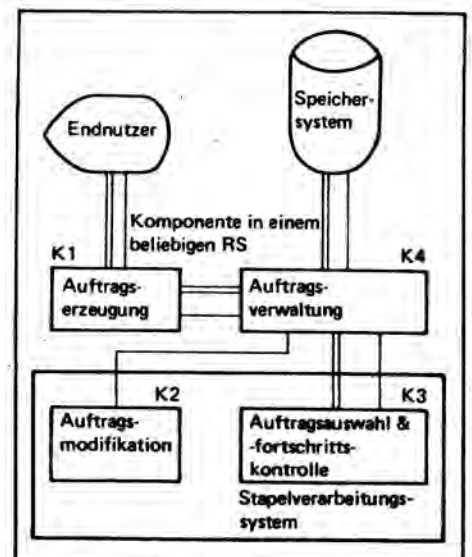
- die Fortschreibung des Bearbeitungsstatus der vom Anwendungsdienst AT verwalteten Aufträge, und

- die Eliminierung von Aufträgen aus dem Anwendungsdienst.

Diese Verwaltungsfunktion, die im wesentlichen das Speichern, Wiederauffinden, Modifizieren und Löschen auftragsrelevanter Daten beinhaltet, ist zweckmäßig unter Nutzung eines Speichersystems zu realisieren.

Die Kommunikationsbeziehungen der vier Komponenten, die in ihrer Gesamt-

Abb. 1 Komponenten zur Realisierung des Anwendungsdienstes Auftragstransfer



heit den Anwendungsdienst AT erbringen, veranschaulicht Abb. 1.

Kommunikationsbeziehungen zur Problemumgebung

In Tab. 1 sind die Interfaces des Anwendungsdienstes AT zur Problemumgebung zusammengefaßt und bezüglich der zu erfüllenden Teilfunktionen weiter detailliert. Die den Teilfunktionen zugeordneten Bezeichner sind als Synonym für den jeweiligen Dienstnutzerprozeß in der Problemumgebung zu verstehen.

Die Zuordnung der Interfaces und der Dienstnutzerprozesse zu den vier Komponenten ist aus Tab. 2 ersichtlich. Diese Tabelle zeigt, daß die Komponente K4 explizit über kein Interface zur Problemumgebung verfügt. Sie kommuniziert endsystemintern mit dem Speichersystem.

Kommunikationsprinzipien

Die Kommunikationsbeziehungen zwischen den Komponenten, die in der ESER-Implementierung endsystemintern ablaufen, sind mit dem Ziel der Übertragbarkeit einzelner Komponenten auf andere Endsysteme, folglich einer dynamischen, endsystemabhängigen Konfigurierung des Anwendungsdienstes AT, analog den Prinzipien der endsystemübergreifenden Kommunikation organisiert.

Die zwischen den Komponenten stattfindende Kommunikation erfolgt mittels AT-Anwendungsprotokoll unter Benutzung der Dienste der unterlagerten Schichten laut OSI-RM. Dieses Protokoll basiert auf dem *Invocation-Send-Prinzip* /3/, d. h., ein die Kommunikation initiiierender Prozeß verharret nach dem Anstoß des Datenaustausches solange im Wartezustand, bis der Partnerprozeß die ankommenden Daten in einem Teilprozeß verarbeitet und mittels Quittung den initiiierenden Prozeß über das Verarbeitungsergebnis informiert und ihn zugleich wieder aktiviert. Für die Abwicklung des AT-Anwendungsprotokolls wurde die Empfehlung X.217 der CCITT /4/ umgesetzt.

Bearbeitungszustände eines Auftrages im Anwendungssystem

Ein beliebiger Auftrag $A_i ::= \{J_{ik}\}$, der dem Anwendungsdienst übergeben wird, durchläuft während der Bearbeitungszeit folgende Zustände:

BATCH

Der Auftrag A_i ist im Anwendungssystem registriert und kann durch ein Stapelverarbeitungssystem zur Verarbeitung ausgewählt werden.

RUN

Der Auftrag A_i wird durch ein Stapelverarbeitungssystem bearbeitet. RUN kennzeichnet dabei einen Komplexzustand, der sich weiter detaillieren läßt.

END

Der Auftrag A_i ist vollständig bearbeitet worden.

Die Elemente J_{ik} (Jobs) des im Zustand RUN befindlichen Auftrages A_i durchlaufen ihrerseits folgende Zustände, die sich aus den Verarbeitungsprinzipien des Betriebssystems OS/ES ergeben:

START

Job J_{ik} steht für Systemeingabe bereit

INQE

Job J_{ik} wurde in eine aktive Systemeingabekette eingereiht

ACTIV

Ein Verarbeitungsschritt des Jobs J_{ik} befindet sich in Bearbeitung

OUTQE

Für Job J_{ik} ist eine Ausgabe erzeugt worden, die für den Endnutzer im Speichersystem bereitzustellen ist

STOP

Job J_{ik} ist vollständig bearbeitet

Der Zustand RUN wird verlassen, wenn alle Jobs J_{ik} des Auftrages A_i in den Zustand STOP übergegangen sind. An dieser Stelle wird wieder das Niveau der Bearbeitung auf Auftragsebene erreicht.

Speichersystem

Die Struktur des Speichersystems unterliegt folgenden Prämissen.

P1: Das Speichersystem ist hierarchisch strukturiert.

– In einer Auftragssteuerdatei (ASD) werden alle AT-spezifischen Informa-

Tab. 1 Interfaces, Funktionszuordnung und Dienstnutzerprozesse des Anwendungsdienstes Auftragstransfer

Interface	Teilfunktionen	Bezeichner
Endnutzer	– Einbringen/Absetzen eines Auftrages an ein beliebiges/bestimmtes Stapelverarbeitungssystem	ASUBMIT/ASUB
	– Statusabfrage bezüglich der vom Endnutzer abgesetzten Aufträge	AINFOR/AI
	– Anzeigen von Verarbeitungsergebnissen eines Auftrages	ALIST
	– Entfernen eines Auftrages aus dem Anwendungsdienst AT	AEND
Bediener	– zwangsweise Beendigung eines Auftrages	ACANCEL/AC
	– Rücksetzen von Aufträgen in den Zustand vor Bearbeitungsbeginn	ARESET/AR
	– Anzeige von Aufträgen mit definierten Bearbeitungsstatus	ADISPLAY/AD
maschinelles Interface	– Auswahl eines Auftrages zur Verarbeitung	ASELECT/AS
	– Registrieren von jobbezogenen Ergebnissen auf OS/ES-Betriebssystemniveau wie:	INIT
	• Jobstart und -stop • Stepstart, -wechsel und -stop	RDR WTR
	– Kumulierung dieser jobbezogenen Ereignisse zu auftragsbezogenen Ereignissen wie:	
	• Auftragstart • Auftragsstop	

Interface	Teilfunktion	zugeordnete Komponente
Endnutzer	ASUBMIT AINFOR ALIST AEND	Auftrags- erzeugung K1
Bediener	ACANCEL ARESET ADISPLAY	Auftragsmo- difikation K2
maschinelle Interface	ASELECT RDR INIT WTR	Auftrags- auswahl und -fortschritts- kontrolle K3
-	-	Auftrags- verwaltung K4

Tab. 2 Zusammenhang zwischen Interfaces, Teilfunktionen und Komponenten der Anwendung Auftragstransfer

tionen zu Aufträgen und deren Jobs verwaltet.

- Je Auftrag werden die notwendigen Eingabedaten (Jobstromfolge) in einer Auftragsdatei (AD) und wahlweise die Ausgabedaten (Protokolle) in einer Auftragsausgabedatei (AOD) kumuliert. ADi und AODi sind über die Informationen zum Auftrag Ai in der ASD erreichbar.

P2: Die das Speichersystem verwaltende Komponente K4 sichert die Konsistenz der Daten eines Auftrages

- bei der parallelen Bearbeitung verschiedener AT-spezifischer Prozesse innerhalb eines RS sowie

- bei gemeinsamer Benutzung des Speichersystems durch mehrere (zwei) RS, über Ressourcenschutzmechanismen /5/.

P3: Das Speichersystem ist auf einer Menge von (gemeinsam nutzbaren) Direktzugriffsspeichern, die nicht zwingend gleichen Gerätetypen sind, anzuordnen. Die Verwaltung des Speichersystems ist nach den E/A-Prinzipien der Datenverwaltung des Betriebssystems OS/ES unter effektiver Ausnutzung der Ressource Speicherplatz organisiert /6/.

Implementierung auf ESER-Rechnersystemen

Der Anwendungsdienst AT wurde in einer ersten Modelllösung für ESER-RS implementiert, die physisch über gemeinsam genutzte Direktzugriffsspeicher, auf denen das Speichersystem residiert, verbunden sind.

Damit erhält das Speichersystem eine Doppelfunktion, es ist:

- lokales Speichersystem der auf jedem RS lokalisierten Komponente K4 und
- globales Kommunikationssystem zwischen den zusammenwirkenden RS.

Der Anwendungsdienst AT wird durch eine eigenständige Aufgabe repräsentiert, die unter Steuerung des Betriebssystems OS/ES die vier definierten Komponenten als Subprozesse verwaltet. Die Initialisierung, Deaktivierung und parallele Steuerung der Subprozesse erfolgt abhängig von den in der Problemumgebung initiierten Dienstnutzerprozessen durch einen weiteren, in verteilten Anwendungen notwendigen Prozeß, dem die Funktion der Systemverwaltung zugeordnet wird.

Die den Anwendungsdienst AT repräsentierende Aufgabe entsteht als Produkt einer Generierung, in der die steuerblockmäßig beschriebenen und zu unterstützenden Komponenten auszuwählen und mit weiteren Steuerblöcken assembliert und in einem Programmverbindenlauf mit den zum Anwendungsdienst zählenden Lademodulen verbunden werden. Für den Aufruf dieses Programmes ist eine Prozedur bereitzustellen, die durch ein Start-Kommando aktiviert werden kann.

Die Kommunikation zur Problemumgebung hin erfolgt über eine Prozeßschnittstelle /3/, die den alternierenden Datenaustausch über einem vom Betriebssystem verwalteten Speicherbereich (SQA) unterstützt. Damit wird einer dynamischen Anzahl von Dienstnutzerprozessen der parallele Zugang zum Anwendungsdienst AT ermöglicht. Als Dienstnutzerprozesse fungieren:

- TSO-Kommandoverarbeiter (ASUBMIT, AINFOR, ALIST, AEND) bzw. äquivalente Stapelverarbeitungsprogramme,

- OS/ES-Bedienerkommandos (ASELECT, ACANCEL, ARESET, ADISPLAY) und

- Rahmenprogramme bzw. Anhangsroutinen für die Systemaufgaben RDR, INIT und WTR.

Die Prozeßkommunikation wird mittels eines speziellen für den Anwendungsdienst AT entwickelten SVC realisiert, der im Rahmen einer Systemanpassung in den Nucleus einzubinden ist.

Die Kommunikation zwischen den Partnerkomponenten (Subprozessen) des Anwendungsdienstes AT erfolgt aufgabenintern über Benutzung der Dienste der unterliegenden Sitzungsschicht. Die Schnittstelle zwischen diesen benachbarten Schichten ist als Programm-/Prozedurschnittstelle /3/ ausgebildet. Damit erfährt das im Anwendungsprotokoll verwirklichte Invocation-Send-Prinzip eine adäquate Umsetzung.

Der Anwendungsdienst Auftragstransfer wurde von einem betrieblichen Themenkollektiv realisiert. Die Implementierung der verteilten Anwendung Auftragstransfer ist in der Assemblersprache unter konsequenter Anwendung der Makros der strukturierten Programmierung realisiert.

Das Gesamtsystem besteht aus 130 Programmmodulen und umfaßt etwa 25 000 Anweisungen.

Der Anwendungsdienst AT befindet sich seit Anfang 1988 auf zwei über gemeinsam nutzbare Direktzugriffsspeicher verbundenen ESER-RS in der praktischen Anwendung.

Erweiterungsmöglichkeiten

Der am *Open-Ended-Design* orientierte Lösungsansatz erlaubt für zukünftige Entwicklungsstufen sowohl

- die funktionelle Erweiterung des Anwendungsdienstes (z. B. Automatisierung der Auftragsauswahl, Qualifizierung der Auftragsplanung),

- die Einbeziehung neuer Anwendungsdienste (Filetransfer) wie auch

- die Übertragung von Komponenten des Anwendungsdienstes AT auf andere, unter Umständen heterogene, RVS bei entsprechendem Ausbau der kommunikationsrelevanten Dienste.

Gegenwärtig wird der Anwendungs-

DAFEMA-PC: Kopplungssoftware für die verteilte Verarbeitung

Dr. Peter Erward
VEB DVZ Magdeburg

dienst funktionell erweitert. Hierzu zählt:

– die Automatisierung der Auftragsauswahl für eine spezielle Auftragskategorie und

– die Ermöglichung der selektiven Jobwiederholung im Rahmen der Bearbeitung eines Auftrages.

Für die Zukunft ist der Ausbau der kommunikationsrelevanten Dienste vorgesehen, um mehreren RS (ESER, PC/BC, etc.) den Zugang zum Anwendungsdienst zu ermöglichen.

Literatur

/1/ ISO, International Standard 7498: Information Processin Systems – Open Systems Interconnection – Basic Reference Model. ISO, Okt./1984

/2/ ISO, Draft Proposal 8831: Information Processin Systems – Open Systems Interconnection – Job Transfer and Manipulation Concepts and Services. ISO, Febr./1985

/3/ Nehmer, J.: Softwaretechnik für verteilte Systeme. Springer Verlag, Berlin, Heidelberg, New York, 1985

/4/ CCITT Recommendation X.217: Common Application Service Elements Definition of Systems Interconnection for CCITT Application. CCITT, 1984

/5/ ESER-Systemunterlagendokumentation, OS/ES Ausgabe 7 – SVS: Hinweise für die Systemanwendung. C 5113-0806-1, VEB Robotron Projekt Dresden, 1987

/6/ OS Direct Access Device Space Management (DADSM). IBM Program Logic, Release 21.7, File No. S 360/370-30, 1976

Das im VEB DVZ Magdeburg entwickelte Datenbankbetriebssystem DAFEMA /1/ (Datenbankbetriebssystem mit Fernzugriff Magdeburg) gehört mit gegenwärtig mehr als 60 nachnutzenden Betrieben in allen volkswirtschaftlichen Bereichen der DDR zu den meist genutzten und zentral empfohlenen ESER-Datenbankbetriebssystemen in der DDR.

DAFEMA ist ein Datenbankbetriebssystem, das gezielte Anfragen und Änderungen im Sekundenbereich realisiert. Charakteristisch für DAFEMA sind folgende Anwendungen:

– Auskunftssystem für Bezirksfernmeldeämter über Telefon- und Vorwahlnummern,

– die Verwaltung und Druckvorbereitung für alle Telefonbücher der DDR,

– Produktionssteuerungssysteme im 24-Stundenbetrieb im Maschinenbau,

– Auskunfts- und Steuerungssysteme in der chemischen Industrie,

– Leitungsinformationssysteme im Bauwesen,

– Systeme zur Abrechnung und Steuerung der Warenzirkulation im Handel.

Im folgenden werden einige wesentliche Leistungsmerkmale des DBBS DAFEMA aufgeführt:

• Die Daten werden relational (satzweise) in einer platzsparenden Form (30 Prozent Einsparung gegenüber der herkömmlichen Form) gespeichert. Die Daten befinden sich in bis zu 64 Basisdateien auf die über Suchdateien (maximal 256) zugegriffen wird, die Schlüssel und physische Satzadresse enthalten.

• Der Nutzer kann über eine prozedurale Endnutzersprache (DML) die Definition und den Aufbau der Datenbasis sowohl im Stapelbetrieb als auch im Dialogbetrieb durchführen. Die Datenbankarbeit mit den wesentlichen Funktionen *Wiederauffinden* (Recherchieren) und *Ändern* kann ebenfalls im Stapel und Dialog erfolgen. Diese Arbeitsweise wird durch eine komfortable Prozedurunterstützung zur oftmaligen Realisierung von komplizierten Anweisungen oder Anweisungsfolgen und durch eine Vielzahl von Organisationsanweisungen zur effektiven Datenbankarbeit zusätzlich unterstützt.

• Der Nutzer kann die DML durch eigene Anweisungen in Form von Anwenderprogrammen in den Sprachen Assembler, PL/1 und C erweitern. Der Zugriff auf die Datenbasis wird dann über in diese Anwenderprogramme eingebettete Zugriffsmakros für Datenbankdateien realisiert. Diese Makros werden mit DAFEMA geliefert.

• Zur Unterstützung von Erfassungsproblemen und Endnutzern in den Fachbereichen wird ein komfortabler Service über die Menü- und Schablonentechnik angeboten.

• Der implementierte Zugriffsschutz läßt den kombinierten Schutz auf Kommandoebene sowie der gesamten Datenbank bis auf Datenfeldebene zu und genügt internationalen Ansprüchen.

Weiterhin wird mit der stellungsorientierten Abspeicherung von Verbundstrukturen dem Nutzer die effektive Unterstützung von produktionsvorbereitenden und -durchführenden Prozessen mit DAFEMA ermöglicht. Es sind DAFEMA-Datenbanken realisierbar, die über die flexible dialoge Arbeit hinaus spezifische technische und technologische Manipulationen erlauben. Diese Funktionen werden durch die Komponenten

– Stapel-Stücklistenprozessor,
– dialoger Stücklistenprozessor wirkungsvoll ergänzt und

– zur Manipulation von Objekten werden DAFEMA-Anweisungen bereitgestellt, die im Dialog und Stapelbetrieb anwendbar sind. Zum Einspeichern, Auslagern und Löschen kompletter Objekte stehen ebenfalls entsprechende Anweisungen für den Nutzer zur Verfügung.

Referenzbetriebe für DAFEMA sind
– VEB IFA-Kombinat PKW Karl-Marx-Stadt,
– VEB Strömungsmaschinen Pirna,
– VEB Rohkombinat Riesa,
– VEB Meßgerätewerk „Erich Weinert“ Magdeburg,

– VEB Elektronische Bauelemente „Carl von Ossietzky“ Teltow,
– Deutsche Post – Institut für Post- und Fernmeldewesen Berlin.

Die große Nutzerakzeptanz von DAFEMA ist das Ergebnis einer kontinu-

♦ Veranstaltung

18. Jahrestagung Grundlagen der Modellierung und Simulationstechnik

Termin: 12. bis 14. 12. 1989

Ort: Rostock

Schwerpunkte:

- Modellierung und Simulation in der Informatik
- Modellierung und Simulation in der Ökonomie
- Methoden der Modellierung kontinuierlicher Systeme
- Grundlagen der Modellierung und Applikation
- Rechen-technische Unterstützung für Entscheidungshilfen

Organisatorische Hinweise: Anfragen sind zu richten an die KDT, WGMA, Clara-Zetkin-Str. 115/117, PSF 1315, Berlin, 1086

ierlichen Weiterentwicklung dieser Basissoftware, in der sowohl vielfältige Anwenderanforderungen aus der praktischen Nutzung als auch der Entwicklungstrend von Hard- und Software ständig Berücksichtigung finden. Dies trifft auch für die hier vorgestellte Basissoftware DAFEMA-PC zu, mit der eine umfassende Integration von Personalcomputern (PC) in die Datenbankarbeit realisiert wird.

Leistungsumfang

Ein wesentliches DAFEMA-Leistungsmerkmal ist die Möglichkeit der Datenfernverarbeitung im Bildschirmdialog, mit der eine arbeitsplatznahe Verfügbarkeit der in der Datenbank verwalteten Massendaten gegeben ist. Mit der forcierten Bereitstellung von Personalcomputern stellt sich die Aufgabe, innerhalb der Datenfernverarbeitung die am Arbeitsplatz verfügbare Computerleistung effektiv in enger Verbindung mit einer zentralen DAFEMA-Datenbank zu nutzen. DAFEMA-PC (asynchron - AP64-Version) ist eine Kopplungssoftware für den Dialog von Nutzern mit dem DBBS DAFEMA mittels Personalcomputer des Typs PC 1715 sowie Bürocomputer A 5120/30 unter dem Betriebssystem CP. Der Personalcomputer ist dabei über das V.24-Interface und die übliche Technik für die Datenfernverarbeitung mit der zentralen ESER-Datenbank verbunden.

DAFEMA-PC realisiert über einen

DFÜ-Treiber im Asynchronmodus (EM62DAF) die für die Dialogarbeit mit DAFEMA benötigten Schreib- und Lesekommandos der AP64-Prozedur. Während einer DAFEMA-Sitzung emuliert das Programm simultan drei AP64-Abonnentenpunkte (Abb. 1). Nutzer Nummer 1 ist ein AP64-Nutzer, welcher den Personalcomputer als Dialogterminal einsetzt. Nutzer Nummer 2 erstellt vor der Sitzung mit Hilfe von Softwarewerkzeugen, wie dem Textprozessor TP oder dem Datenbankbetriebssystem für Personalcomputer REDABAS, eine Kommandodatei, in die er alle von ihm gewünschten, durch DAFEMA abzuarbeitenden Anweisungen schreibt. Während der DAFEMA-Sitzung mit DAFEMA-PC werden diese Anweisungen im Hintergrund zum Bildschirmdialog des Nutzers Nummer 1 an DAFEMA übertragen, dort abgearbeitet und das Ergebnis durch DAFEMA-PC in eine Diskettendatei abgelegt. Diese Ergebnisse können dann nach der Sitzung, offline vom Nutzer Nummer 2 mit Hilfe von DAFEMA-PC, TP oder REDABAS ausgewertet werden. Neben dem *Bildschirmnutzer* und dem *Diskettennutzer* stellt DAFEMA-PC noch einen Drucker als drittes DAFEMA-Terminal bereit (Nutzer Nummer 3).

Abfassung und Auswertung der Nachrichten des Diskettennutzers werden somit von der eigentlichen Sitzung getrennt und gehen damit nicht zu Kosten der DFV. Zusammen mit dem Simultanbetrieb zweier Nutzer über eine Leitung führt das zu einer optimierten Auslastung der beteiligten DFV-Ressourcen. Ein weiteres Leistungsangebot von DAFEMA-PC ist die Möglichkeit, DAFEMA-Dateiauszüge am PC bereitzustellen und diese dort dezentral zu verwalten. Diese Dateiauszüge werden durch DAFEMA-PC im REDABAS-Dateiformat (DBD-Dateien) bereitgestellt.

Dialogarbeit mit DAFEMA-PC

Bildschirmdialog

Die DFV-Abarbeitung von DAFEMA unter TCAM-Steuerung wird über ein zentrales DFV-Steuerprogramm (ZNSP)

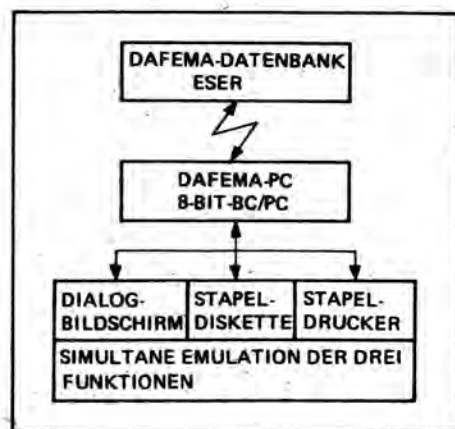
/2/ auf der Basis von TCAM abgewickelt. Als Terminal fungieren im Falle von DAFEMA-PC die Konsole (Bildschirm, Tastatur), das Diskettenlaufwerk und der Drucker. Für diese ist jeweils eine TERMINAL-Eintragung durch Kodieren des TERMINAL-Markos erforderlich. Das Diskettenlaufwerk wird aus DAFEMA-Sicht wie ein Bildschirm behandelt und ist deshalb auch als solcher anzugeben. Die TERMINAL-Eintragung wird neben der Beschreibung des Gerätetyps zur Adressierung genutzt. Dazu ist im ADDR-Operanden die physische Adresse anzugeben.

Für die Realisierung ist auf dem PC das V.24-Interface und ein lauffähiges SCP-Betriebssystem notwendig. Weiterhin ist der DFÜ-Treiber EM62DAF notwendig. Er ist ein Emulatorprogramm zur Umsetzung der *Datenübertragungsprozedur zur zeichenweisen asynchronen Datenübertragung im Unterordnungsbetrieb in die CP/M-Ein-/Ausgabekanäle "RDR:" und "PUN:"*

Es basiert auf einer vom PCK Schwedt vertriebenen Software, welche an die DAFEMA-Spezifik des Programms DAFEMA-PC angepaßt wurde. Als Parameter sind für den DFÜ-Treiber die Steuergeräte- und die Bildschirmadresse notwendig, falls andere als die generierten Standardwerte gewünscht werden. Die Kommunikation zwischen Nutzer und dem Emulatorprogramm erfolgt über einen Kopfblock, der Informationen über die Gerätekonfiguration, die Übertragungstechnik und die verwendeten Adressen, für den der Treiber generiert ist, enthält.

DAFEMA-PC setzt eine betriebsbereite Übertragungsstrecke voraus. ESER-seitig kann das DBBS DAFEMA nach dem dezentralen Start von DAFEMA-PC gestartet werden. Arbeitet ZNSP/2/im gemischten Betrieb mit TSO, so ist dieser DAFEMA-Start auch vom PC aus möglich. Dem Nutzer steht ein in der Kurssteuerung TP-analoger Fullscreen-Editor zur Verfügung, mit dessen Hilfe er die abzusendenden Nachrichten editiert. Analog der Emulationssoftware EM62 sind diese in SOM (#) und EOM (!) einzuschließen. Der DA-

Abb. 1 Funktionen von DAFEMA-PC



FEMA-Dialog wird dabei durch die Laufwerke und den Drucker unterstützt. So ist es zum Beispiel möglich, Routineanweisungen auf der Diskette zu sammeln, und diese bedarfsweise prozedurartig auf den Bildschirm zu holen. Nach Ausfüllen eventuell vorhandener Leerfelder werden sie an DAFEMA abgesendet. Andererseits hat der Nutzer die Möglichkeit, ihn interessierende Details einer DAFEMA-Antwort durch Markieren als Sreencopy auf Drucker oder Diskette abzulegen.

Eine spezielle Funktion erfüllt die letzte Bildschirmzeile. Sie dient als Nachrichten- und Kommandozeile. In ihr werden Fehlermeldungen und Mitteilungen von DAFEMA, ZNSP und DAFEMA-PC sowie nutzerdefinierte Mitteilungen des *Diskettennutzers* angezeigt. Über diese Bildschirmzeile kommunizieren der *Bildschirmnutzer* und der *Diskettennutzer* miteinander. Wegen des prinzipiell unabhängigen und asynchronen Laufes der Dialoge dieser beiden Nutzer stoppt eine Anzeige in dieser Zeile den Dialog des *Bildschirmnutzers* und fordert von ihm nach deren Auswertung zumindest eine Quittierung. Danach steht ihm der zuvor gerettete ursprüngliche Inhalt wieder zur Verfügung, und er kann seinen Dialog fortsetzen.

Diskettennutzer

Zur Anmeldung und Steuerung des Sitzungsverlaufes des *Diskettennutzers* wird mittels der Escape-Taste in den sogenannten Kommandomodus umgeschaltet. Der Nutzer wird dann in der Nachrichten- und Kommandozeile zur Eingabe seines speziellen Kommandos aufgefordert. Die folgenden Kommandos sind möglich:

- D)rive
- Setzen Standardlaufwerk
- F)rom
- Anmelden DAFEMA-Kommandofile
- G)o/W)ait
- Starten/Stoppen des DAFEMA-Kommandofiles
- T)o
- Generieren File zur Aufnahme von Ergebnissen
- L)ist

```
.DM BEGINN DER DISKETTENARBEIT
/1/ *TCAMON (DAFEMA,1234)!
DIES IST EIN DEMONSTRATIONS-
BEISPIEL
.FI USER.COD CHANGE
.DM DISKETTE ANGEMELDET
/2/ *HLP!
/3/ *R $B1!
ES FOLGT EIN SAVE
.ES S
DAS ERSTE KOMMANDOFILE IST
ABGEARBEITET
EINFUEGEN DES KOMMANDOFILES
DAFSEN2.TXT IN DIE ABARBEITUNG
MIT AUFFORDERUNG ZUM
DISKETTENWECHSEL
.FI DAFSEN2.TXT CHANGE
ENDEMITTEILUNG FUER
DAFSEN2.TXT
.DM RECHERCHEN FUER
ABTEILUNG 2 ABGEARBEITET
/4/ *LIS!
/5/ *R $B1 'D(F1)!'
.DM FELD1 ALLER DATENSAETZE
ERHALTEN *** EMPFEHLE SAVE
/6/ *HLP RI
/7/ *R $B1
'A(MAX=1),D(F1-F5)!'
.ES S
PROGRAMMIERTES SAVE UND
ENDEMITTEILUNG
.DM BIN FERTIG ***
```

Abb. 2 Beispiel für eine Kommandodatei

Auflisten von Files

- S)ave
- Sichern des Ergebnisfiles
- P)rot
- Protokoll und Statusinformationen
- C)opy
- Bildschirmdaten auf Drucker oder Diskette
- R)repeat
- Wiederholen einer LAD-SDF-Anweisung
- ?
- Auflisten der möglichen Kommandos. Eingegeben wird der hervorgehobene Kennbuchstabe in Groß- oder Kleinschreibung. Nach Abarbeitung oder Abbruch eines Kommandos wird selbständig in den Dialogmodus zurückgekehrt. Eine abzuarbeitende DAFEMA-Kommandodatei wird mittels FROM angemeldet und dann durch GO gestartet.

GO bewirkt, daß die nächste anstehende Anweisung aus der Kommandodatei an DAFEMA gesendet wird. Der weitere Ablauf sieht dann so aus, daß auf die Antwort gewartet, diese bei Eintreffen in die mittels TO generierte Datei abgespeichert und danach selbständig die nächste Anweisung abgesendet wird. Jede eintreffende Antwort bewirkt das Absenden einer neuen Anweisung. Nach Abarbeitung der Anweisung aus der DAFEMA-Kommandodatei wird die Ergebnisdatei mittels SAVE geschlossen und steht für die weitere Auswertung etwa mittels LIST oder nachfolgend aufgerufener Werkzeuge (z. B. TP) zur Verfügung. Neben dieser Anzeigefunktion des Sitzungsergebnisses dient das LIST-Kommando der Unterstützung des Dialogs des *Bildschirmnutzers*.

Die Kommandodatei ist eine durch CR/LF strukturierte Textdatei. Sie kann mit einem beliebigen Texteditor erstellt werden. Verwendet werden können zum Beispiel:

- TP, Option N,
- REDABAS, Kommando MODIFY COMMAND,
- Turbo-PASCAL, Kommando EDIT.

Die Kommandodatei wird durch SO ('#') und EOM (!) in einen Kommentarteil, welcher nicht gesendet wird, und einen Nachrichtenteil, welcher gesendet wird, gegliedert. Die zu sendenden DAFEMA-Anweisungen sind alle in SOM und EOM einzuschließen. Abb. 2 enthält ein Beispiel einer Kommandodatei. Die erste abgesendete Nachricht ist das TCAMON-Kommando, gefolgt von der COD-Anweisung. Der Text davor bzw. dazwischen ist Kommentartext. Mit Hilfe von Kommentaren wurden hier z. B. die Anweisungen durchnummeriert, so daß bei der Abarbeitung mittels des Kommandos LIST sofort die korrespondierenden Anweisungen zu der in der Protokollfunktion PROT ausgewiesenen Nummer gefunden werden können. Eine weitere Nutzungsmöglichkeit der Kommentare besteht in der Verwendung von sogenannten *Dot-Kommandos*. Dot-Kommandos sind eine spezielle Form von Kommentaren, die den

Sitzungsverlauf des *Diskettennutzers* steuern. Sie sind (bis auf ES) mit denen des Textprozessors voll verträglich und haben beim Ausdruck der Kommandodatei mit KOMBO-DRUCK eine analoge Wirkung. Folgende Dot-Kommandos werden unterstützt:

.FI (dateispez)

An diese Stelle wird beim Senden die spezifizierte Datei eingefügt und gesen-

.DM (hilfstext)

Der Hilfstext wird auf dem Bildschirm angezeigt.

.ES (kennbuchstabe)

Das entsprechende Kommando des Kommandomodus wird ausgeführt.

Dot-Kommandos beginnen mit einem Punkt (dot) in der ersten Spalte, gefolgt vom Kommando und etwaigen Parametern. Sie müssen allein auf einer Zeile stehen und dürfen nur in den Kommentarteilen der Kommandodatei (d. h. nicht in den durch SOM und EOM mar-

kierten DAFEMA-Anweisungen) angegeben werden.

Durch ein .FI-Kommando wird die Abarbeitung in der angegebenen Datei fortgesetzt. Mittels dieses Kommandos ist es also möglich, ein hierarchisches Konzept der Dialogarbeit des *Diskettennutzers* zu verwirklichen. Die mittels FROM angemeldete Kommandodatei wirkt dabei als reine Steuerdatei, die die einzelnen, tatsächlich abzuarbeitenden DAFEMA-Kommandodateien aufruft. Der nach dem .DM-Kommando folgende Text wird in der maximalen Länge einer Bildschirmzeile in der Nachrichten- und Kommandozeile angezeigt. Nach einer Quittierung wird die Abarbeitung der Kommandodatei fortgesetzt. Da vor der Quittierung der Kommandomodus aufgerufen werden kann, sollte das .DM-Kommando dazu verwendet werden, den Bediener am Bildschirm über den Fortschritt des *Diskettendialogs* zu informieren, ihn zu be-

stimmten Kommandos des Kommandomodus aufzufordern oder einen Diskettenwechsel zu ermöglichen.

Das .ES-Kommando ist das mächtigste Dot-Kommando und gestattet es dem *Diskettennutzer*, den Sitzungsverlauf flexibel und optimal an seine Dialogaufgaben anzupassen. Alle durch Eingabe eines Kennbuchstabens möglichen Funktionen des Kommandomodus werden auch durch das .ES-Kommando ausgelöst. Der Kennbuchstabe steht dabei hinter .ES.

Wird bei Kommandos, die einen Dateinamen erfordern, lediglich der Kennbuchstabe geschrieben, so wird der Bediener dazu aufgefordert, den Dateinamen einzugeben. Wird der Dateiname jedoch mit in die Steuerdatei geschrieben, so wird der Bediener nicht beansprucht. Abb. 3 enthält ein Beispiel für eine Kommandodatei unter Verwendung von Dot-Kommandos.

REDABAS-Dateiauszüge

Zur Unterstützung der dezentralen Arbeit mit DAFEMA-Datenbeständen enthält DAFEMA-PC auch eine Komponente zur Übertragung von DAFEMA-Dateiauszügen an den PC. Von DAFEMA-PC erzeugte DAFEMA-Dateiauszüge sind REDABAS-Dateien. Sie stehen damit einer unmittelbaren Auswertung durch REDABAS-Nutzer/-Anwenderprogramme zur Verfügung.

Um die Übertragung einer Datei von DAFEMA an DAFEMA-PC einzuleiten, ist an DAFEMA die Anweisung LAD SDF zu senden, die Daten aus der zentralen DAFEMA-Datenbank im Standard-Datenformat auslagert. Das Absenden dieser Anweisung kann sowohl vom Bildschirm als auch aus einer Kommandodatei heraus geschehen.

Eine vom Bildschirm initiierte Dateiübertragung wird über die Diskettenadresse abgewickelt. Dies hat den Vorteil, daß nach Absenden der entsprechenden Anweisung der DAFEMA-Dialog am Bildschirm fortgesetzt werden kann, während im Hintergrund der Dateitransfer läuft.

Eventuelle Fehlermeldungen von DAFEMA betreffs der Dateiauslagerung werden auf dem Bildschirm zur Anzeige

Abb. 3 Beispiel für eine Kommandodatei unter Verwendung von Dot-Kommandos

```
#TCAMON(DAFEMA,T002)!
.FI USER.COD CHANGE
.DM DISKETTE ANGEMELDET
.FI ABSATZ.DAF CHANGE
.FI LAGER1.DAF CHANGE
.DM LAGER1 ABGEARBEITET
.FI LAGER2.DAF
.DM IHRE ZUGRIFFSBERECHTIGUNG BITTE
.FI KADER.COD CHANGE
.FI KADER.DAF CHANGE
.DM IHRE ZUGRIFFSBERECHTIGUNG BITTE
.FI BILANZ.COD CHANGE
.FI BILANZ.DAF
.DM WENN FEHLERMELDUNG "BILANZ.DAF NICHT GEFUNDEN"
App. 629 INFORMIEREN
.ES S
.DM BIN FERTIG ***
mit der Zugriffsberechtigungsdatei USER.COD (mit erläuternden Kommentaren)
.ES T CODANT
Anlegen einer Datei CODANT zur Aufnahme der Antwort
#COD 129!
Senden des Nutzerkodes
.ES S
SAVE von CODANT
.ES L CODANT
Darstellen der Antwort auf dem Bildschirm
BITTE ANWEISUNG EINGEBEN oder COD NICHT VEREINBART
.DM WENN COD OK DANN W... SONST ERNEUT F)rom UPDATES.DAF
Stoppen der Weiterarbeit, um auf die Antwort geeignet zu reagieren
```

NetWare Hochtechnologie bei PC-Netzen

Rolf Stickler, Kombinat Seeverkehr und Hafenwirtschaft Rostock

Axel Wüstemann, WPU Rostock, Institut für sozialistische Wirtschaftsführung

gebracht. Um bei kleineren Syntaxfehlern nicht die ganze Anweisung noch einmal eingeben zu müssen, kann das Repeat-Kommando benutzt werden. Nach Eingabe von 'r'/'R' im Kommandomodus steht die abgesendete Anweisung auf dem Bildschirm zur Korrektur bereit.

Ein weiteres Leistungsangebot von DAFEMA-PC zur Zusammenarbeit mit REDABAS ist ein REDABAS-DAFEMA-Kommandokonverter. Mit seiner Hilfe wird der Nutzer, der mittels REDABAS die durch DAFEMA-PC bereitgestellten Dateiauszüge manipuliert, in die Lage versetzt, diese Änderungen auch im zentralen DAFEMA-System nachzuvollziehen. Dazu werden sämtliche ändernden REDABAS-Kommandos in analoge DAFEMA-Anweisungen umgesetzt und in einer Diskettendatei abgespeichert, die bei einer erneuten DAFEMA-PC-Sitzung als Kommandodatei abgearbeitet werden kann.

Schlußbemerkungen

Auf der Grundlage der im Beitrag vorgestellten Lösung wird vom VEB DVZ Magdeburg eine Version DAFEMA-PC für die synchrone Übertragungsprozedur BSC 3 und die Emulation des Bildschirmgerätes EC 7925 für genannte 8-Bit-Technik bereitgestellt. Darüber hinaus wird gegenwärtig die Integration der 16-Bit-Arbeitsplatzrechner EC 1834 und AC 7150 in die Datenbankarbeit mit DAFEMA vorbereitet.

Bei Nachnutzungsinteresse sind über folgende Adresse weitere Informationen verfügbar:

VEB DVZ Magdeburg, Bereich Absatz und Beschaffung, PSF 312, Magdeburg, 3010; Tel. 59 43 47.

Literatur:

- /1/ Riewendt, P.: DAFEMA-Datenbanksystem für die Fernverarbeitung, rd 17 (1980) 12, S. 24-25
- /2/ Anwendungsdokumentation ZNSP Zentrales DFV-Steuerprogramm Version 3. VEB DVZ Magdeburg 1985

NetWare ist ein Netzwerkbetriebssystem für Netze von Personalcomputern (PC-Netze). Es wurde 1982 durch die amerikanische Firma Novell auf dem Markt eingeführt und hält mit 30,8 Prozent /1/ den weitaus größten Anteil am Netzbetriebssystem-Markt für Personalcomputer.

NetWare soll geeignet sein, als kostengünstigere und flexiblere Alternative zu Mehrplatzsystemen eingesetzt zu werden. In /1/ wird eine Vergleichsrechnung vorgenommen, nach der ein PC-Netz etwa nur die Hälfte eines Mini-Rechners mit einer entsprechenden Anzahl von Terminals kostet. Auch im Zeitverhalten schneidet ein PC-Netz wie NetWare besser ab. Mit größer werdender Anzahl von Terminals nimmt in einem Mehrbenutzersystem (P8000, K1630) die Zugriffszeit erheblich zu.

Für die geschilderte Einsatzumgebung sind vier Eigenschaften des Netzbetriebssystems von besonderer Bedeutung:

- Geringe Antwortzeiten, insbesondere beim Zugriff auf Netzwerkressourcen durch mehrere Nutzer gleichzeitig.
- Die Integrationsfähigkeit des Netzbetriebssystems in übergeordnete Rechnerkommunikations-Strukturen.
- Eine vielfältigen Bedürfnissen entsprechende Benutzeroberfläche, die eine individuelle Zuordnung der für einen Nutzer notwendigen und erlaubten Netzwerk-Ressourcen gestattet (Datenschutz).
- Hohe Zuverlässigkeit, insbesondere der zentralen Komponenten, der Server. Novell war beim Entwurf des Netzbetriebssystems bestrebt, diesen Leistungsanforderungen durch Einsatz modernster Software-Konzepte zu entsprechen.

Die Architektur von NetWare

Novell versteht sich in seiner Rolle als führender Hersteller von Netzbetriebssystemen vor allem auch als Technologie-Wegweiser. Diese wegweisende Rolle kommt in den NetWare zugrunde gelegten Architektur-Prinzipien zum Ausdruck. Diese Prinzipien werden unter dem Begriff *UNA: Universal Net-*

Ware Architecture zusammengefaßt und umfassen

- die Fileserver-Umgebung,
- die OPT: Open Protocol Technology,
- die SFT: System Fault Tolerance und
- die Netzwerk-Verwaltung.

Das Netzbetriebssystem besteht aus den Komponenten

- Einplatzbetriebssystem (MS-DOS ab Version 2.0)
- die DOS-Shell-Schnittstelle
- den Fileserver - entweder als spezielles Gerät oder unter MS-DOS ladbare Software
- Dienst- und Hilfsprogramme.

Die Konzeption ist hardwareunabhängig. Es werden LAN aller wesentlichen Hersteller unterstützt. Novell vertreibt das System auch als OEM-Software an Dritthersteller.

Abb. 1 zeigt das Zusammenspiel dieser Komponenten von NetWare.

Nachfolgend werden die vier NetWare-Architekturelemente näher dargestellt.

Die Fileserver-Umgebung

Die *Shell* ist die in eine Arbeitsstation zu ladende Komponente von NetWare. Sie realisiert die Kommunikations-Verbindung zwischen Arbeitsstationen und/oder Anwendungen und dem Server. Diese Verbindung erfolgt direkt ohne Nutzung von DOS. Durch die *Shell* wird unabhängig von der tatsächlichen vorhandenen DOS-Version die Kompatibilität von NetWare zu DOS 3.1. erzeugt, wobei es entsprechend der gegebenen DOS-Version verschiedene *Shell*-Versionen gibt. Daneben stellt sie eine Reihe weiterer DOS-Funktionen (INT 21H) bereit.

Fileserver sind das Herz von NetWare. Sie verwalten den Dateizugriff, kontrollieren die Kommunikation zwischen dem Netzwerk zugehörigen PC und steuern die zugehörigen Drucker und Plattensubsysteme. Jeder der Fileserver kontrolliert die Nutzung seiner eigenen Aktivitäten. In einem System mit mehreren Servern koordiniert jeder Server seine Arbeit mit den anderen zugehörigen Servern so, als wenn das gesamte Netzwerk wie *eine wirksame Einheit* arbeitet.

Jedem Fileserver können bis zu fünf

Drucker zugeordnet werden. Diese Drucker werden dann als Netzwerkdrucker bezeichnet. Sie können von allen Netzwerknutzern nur über den Fileserver verwendet werden.

Jeder Fileserver muß mindestens ein oder mehrere Festplatten haben. Auf diesen werden Directories eingerichtet. Die Files eines solchen Directories bilden in der Regel eine logische Einheit.

NetWare können jedem Nutzer bis maximal 26 Directories zugeordnet werden, wobei jedes Directory durch einen lateinischen Buchstaben als virtuelles Laufwerk (auch als Netzwerk-Laufwerk bezeichnet) gekennzeichnet wird.

Hier ist es allerdings sinnvoll, die Laufwerksbezeichnungen, die durch DOS beansprucht werden, nicht einem Netzlaufwerk zuzuordnen. Würde man nach dem Booten eines PC dem Netzwerk auch die Laufwerke A – E zuordnen, gäbe es keine Möglichkeit auf die lokalen Laufwerke des PC zuzugreifen.

Abb. 1 Komponenten von NetWare
Abb. 2 Schnittstellen von NetWare

Die Technologie der offenen Protokolle

Novells Strategie zur Einbindung des Netzbetriebssystems in übergeordnete Kommunikations-Infrastrukturen basiert auf der Technologie der offenen Protokolle (OPT) /5/.

Die Philosophie von OPT besteht darin, die Möglichkeit zu schaffen, verschiedene Arbeitsstationen mit verschiedenen Wirts- und Netzbetriebssystemen (d. h. verschiedenen Protokollen) an einen NetWare-Server anzuschließen. Diese können so dessen Ressourcen in ihrer eigenen, natürlichen Umgebung nutzen.

So wie NetWare hardwareunabhängig konzipiert ist, wird es durch OPT auch unabhängig von Dienst-Protokollen.

Die Grundlage zur Realisierung dieses Konzeptes bildet das *SPG: Service Protocol Gateway* – eine Spezifikation für die Umsetzung der verschiedenen, firmenspezifischen Fileserver-Protokolle auf das *NCA: NetWare Core Protocol*, das NetWare eigene Fileserver-Protokoll.

Weitere Möglichkeiten zur Kopplung

von NetWare mit anderen Systemen sind:

- vollständige Kompatibilität für Anwendungsprogramme, die auf MS-DOS 3.x und NETBIOS basieren und damit zu MS-NET und dem IBM PC Network Program /6/.

- Internetzfähigkeit durch interne (im Server) und externe (als extra Gerät) Brücken

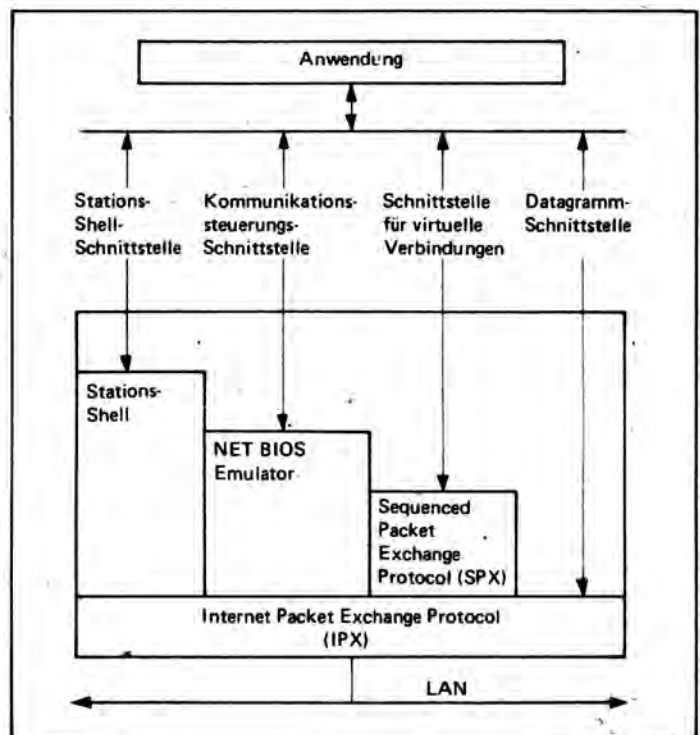
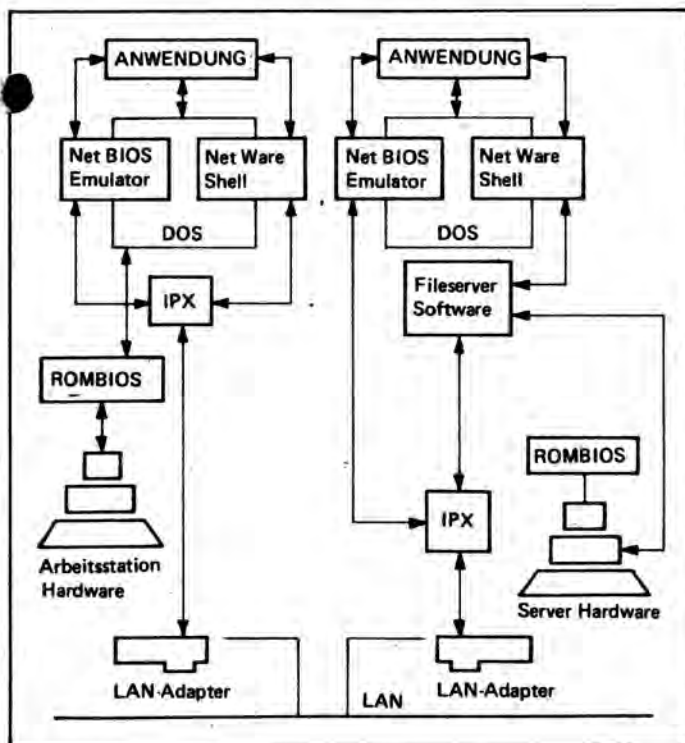
- Anschlußmöglichkeiten und transparentes Einbeziehen von entfernten Arbeitsstationen z. B. über Modems und Postleitungen

- Übergangsmöglichkeiten zu Weitverkehrsnetzen (X.25) und IBM-Großrechnern (SNA)

- Zusammenarbeit mit Minirechnern der Firma Digital Equipment (VAX) durch das Produkt NetWare/VMS.

Die Systemfehler-Toleranz

Die Zuverlässigkeit wird durch bestimmte Zusatzmodule (*SFT: System Fault Tolerance*) erhöht. Es gibt drei Stufen der Fehlertoleranz. Sie reichen von der einfachen Kontrolle gerade ge-



schriebener Daten über das Spiegeln und Duplizieren von Laufwerken bis hin zum Duplizieren des gesamten Fileservers.

Letztendlich kann die Zuverlässigkeit, insbesondere von Datenbankanwendungen durch ein Transaktionsverwaltungssystem (*TTS: Transaction Tracking System*), das ein extra Produkt darstellt, erhöht werden. Mit Hilfe dieses Moduls sind atomare Aktualisierungen von Daten möglich.

Die Netzwerk-Verwaltung

Das Netzwerk-Verwaltungs-System umfaßt die Definition und Einrichtung von Netzwerkressourcen, verschiedene Funktionen, wie Abrechnung, Statistiken usw. sowie ein mehrstufiges, ausgeklügeltes Zugriffsschutz-System auf die verwalteten Datenbestände.

Der grundlegende Nutzen dieses Systems besteht in der Möglichkeit, eine sichere und geschützte Betriebskonfiguration zu schaffen, die auf die individuellen Bedürfnisse jedes Nutzers, jeder Arbeitsgruppe, jeder Abteilung oder des ganzen Betriebes zugeschnitten ist. Verschiedene Abstufungen von freiem und eingeschränktem Zugriff können miteinander kombiniert werden, um so die ideale Sicherheitsumgebung zu bilden.

Arten des Datenschutzes in NetWare

Die Sicherheit von Netzwerkdateien kann nach vier verschiedenen Methoden festgelegt werden:

- LOGIN/Password-Schutz
- Trustee-Schutz
- Directory-Schutz
- File-Attribut-Schutz.

Diese Methoden können getrennt oder in verschiedenen Kombinationen verwendet werden.

① LOGIN/Password-Schutz:

Durch den Fileserver wird geprüft, ob der Nutzer berechtigt ist, am Netzbetrieb teilzunehmen. Jeder Nutzer kann sich durch ein Password, daß von ihm selbst mit einem Dienstprogramm zu warten ist, vor unberechtigtem Zugriff schützen.

② Trustee-Schutz:

Er kontrolliert die verschiedensten Zu-

griffsrechte der einzelnen Nutzer für die einzelnen Directories. Die Zugriffsrechte erhalten die Nutzer vom Supervisor mit dem Programm SYSCON entsprechend ihrer Arbeitsaufgabe.

Zu den acht Rechten gehören:

- Read: geöffnete Datei lesen
- Write: geöffnete Datei schreiben
- Open: Datei öffnen
- Create: neue Datei erstellen
- Delete: Dateien löschen
- Parental: persönliches Verzeichnis verwalten
- Search: im Verzeichnis suchen
- Modify: Datei-Attribute ändern.

Die Trustee-Rechte dehnen sich auch auf die Subdirectories aus.

③ Directory-Schutz:

Für jedes Directory werden die maximal möglichen Trustee-Rechte festgelegt. Diese Rechte gelten nicht für die zugehörigen Subdirectories.

④ File-Attribut-Schutz:

File-Attribute, die sich auf den Dateischutz unter NetWare beziehen, sind R/O (read only), bzw. R/W (read write). Mit R/O können einzelne Dateien gegen Überschreiben oder Löschen geschützt werden.

Effektive Rechte

Zugriffsrechte werden nur dann wirksam, wenn sie sowohl im Trustee-Schutz als auch im Directory-Schutz vergeben worden sind. Die Fileattribute haben eine höhere Priorität als die effektiven Rechte. Das bedeutet, wenn ein Nutzer die Rechte Modify bzw. Delete hat und als Fileattribut Read/Only gesetzt wurde, können die Nutzer-Rechte nicht genutzt werden.

Einige Implementierungsdetails

Wir konzentrieren uns hier nur auf solche Aspekte, die die hohe Leistungsfähigkeit von NetWare bestimmen.

Die wesentlichen Engpässe des Netzbetriebssystems liegen für die Entwickler von NetWare

- im Übertragungskanal,
- im Server-Betriebssystem und
- beim Plattenzugriff.

Die Datenübertragung in NetWare

Die Architektur der Übertragungskanäle ist so ausgelegt, daß sie verschiedenen Nutzerbedürfnissen gerecht wird.

Es stehen vier Schnittstellen zur Verfügung (Abb. 2):

- eine Stations-Shell Schnittstelle,
- eine Schnittstelle auf dem Niveau der Kommunikationssteuerung,
- eine Schnittstelle für virtuelle Verbindungen sowie
- eine Datagramm-Schnittstelle.

Die Shell-Schnittstelle basiert auf dem *NCP: NetWare Core Protocol*, dem NetWare-Datei-Dienst-Protokoll. Das Protokoll ist verbindungsorientiert. Es werden über 100 Protokollfunktionen, die sich zehn Funktionsklassen zuordnen lassen, realisiert.

Die Kommunikationssteuerung-Schnittstelle entspricht dem NetBIOS-Standard von IBM /6/. Sie wurde eingeführt, um Kompatibilität zu entsprechenden Anwendungen herzustellen.

Die Datagramm-Schnittstelle basiert auf dem *IPX: Internet Packet Exchange Protocol* und die Schnittstelle für virtuelle Verbindungen auf dem *SPX: Sequenced Packet Exchange Protocol*. Es ist eine verbindungsorientierte Transportschicht-Schnittstelle.

Beide Protokolle sind Novell-Implementierungen der entsprechenden Protokolle aus der *XNS: Xerox-Network-System-Architektur*. XNS wurde von vornherein für den Einsatz im LAN-Bereich ausgelegt. Es wird hier u. a. angenommen, daß die meisten Übertragungsfehler durch die Sicherungsschicht erkannt werden. Dadurch kann auf aufwendige Fehlererkennungs- und Behebungsmaßnahmen verzichtet werden.

Die anwendungsorientierte Shell setzt direkt auf die niedrigst mögliche, hardwareunabhängige Übertragungsschicht auf. Dadurch wird eine hohe Effektivität bei der Datenübertragung erreicht.

Server-Betriebssystem

Die Basis des Fileservers bildet ein eigenes, für diesen Zweck spezialisiertes Multitasking-Betriebssystem. Er ist also keine eigentliche Anwendung von DOS.

Man ging von der Erkenntnis aus, daß universelle Betriebssysteme, wie MS-DOS oder UNIX letztendlich nicht die geforderten Leistungsparameter eines Fileservers unterstützen. Beim Entwurf des NetWare-Fileserver-Betriebssystems stellte man sich die grundlegenden Ziele

- Schaffung von Voraussetzungen für Zusammenarbeit des Servers mit Arbeitsplatzrechnern, die unterschiedliche Betriebssysteme nutzen. Das betrifft vor allem die Gestaltung der Nutzer/Server-Schnittstelle;
- die Antwortzeiten der Netzwerkdienste müssen mit denen alleinstehender Arbeitsplatzrechner konkurrieren können;
- der Entwurf muß offen sein für die Integration weiterer Dienste, insbesondere solcher anderer Hersteller.

Das Betriebssystem unterstützt u. a. den Protected-Mode beim 80286-Prozessor, wenn die Server-Maschine ein IBM-AT-kompatibler Rechner ist. Der Zugriff auf die Hardware erfolgt unter Umgehung des ROM-BIOS. DOS wird als Task des Server-Betriebssystems abgearbeitet, so daß der Rechner noch als Arbeitsstation nutzbar ist.

abgesetzte Server (extra Gerät) laufen vollständig ohne DOS, mit dem eigentlichen NetWare-Betriebssystem. Novell empfiehlt für mehr als vier Arbeitsstationen, abgesetzte Server zu benutzen. Aber auch bei geringerer Anzahl von Arbeitsstationen sind sie aus funktionaler Sicht günstiger. Ihre Nutzung ist allerdings teurer, da ein extra Gerät angeschafft bzw. ein PC (am besten ein AT) nur für diese Aufgabe abgestellt werden muß.

Einen Eindruck über die Leistungsfähigkeit von Server-Geräten unter NetWare vermittelt Tab. 1.

Es existieren insgesamt sechs verschiedene Server-Anforderungsprozesse, die quasiparallel Aufträge entgegennehmen können.

In einem Server können bis zu vier verschiedene Netzwerkkarten installiert sein, so daß er als Brücke in einem aus vier Teilnetzen bestehenden Internetzwerk fungieren kann.

100	logische Nutzer pro Server
1000	offene Dateien pro Server
5	gepufferte Drucker
256	MByte Daten-Volumen
24000	Verzeichnis-Einträge pro logischem Laufwerk
1500	Cache-Puffer
32	Laufwerke pro Server
2	GByte Massenspeicher-Kapazität

Tab. 1 Maximale Leistungsparameter eines abgesetzten Servers auf der Basis des Prozessors 180286 /2/

Die Plattenzugriffs-Kanäle

Großes Gewicht wurde auf die Optimierung des Plattenzugriffs gelegt, da die meisten Dienstanforderungen direkt damit verbunden sind. In NetWare werden die nachfolgenden Methoden benutzt, um das Leistungsverhalten des Servers beim Plattenzugriff zu verbessern:

① Indizierung der Dateizuordnungstabelle

Die Indizierung der *FAT: File Allocation Table* ermöglicht einen schnelleren wahlfreien Zugriff auf große Dateien. Dadurch dauert das Suchen in einer 100 MByte großen Datei etwa nur halb so lange, wie bei einer nicht indizierten FAT.

② Nutzung von Hash-Algorithmen und Cache-Speichern für ganze Verzeichnisse

Ganze Verzeichnisstrukturen ausgewählter Verzeichnisbereiche werden in den Hauptspeicher eingelesen und bei jeder Änderung fortgeschrieben. Hash-Algorithmen werden genutzt, um einerseits Dateien entsprechend ihrem Verzeichnis- und Unterverzeichnisbereich sowie andererseits um die Verzeichnisstruktur innerhalb jedes Verzeichnisbereiches zu indizieren. Diese beiden Hash-Algorithmen besitzen unterschiedliche Effektivitätsmerkmale.

③ Cache-Speicher für Daten

Der Plattenprozeß ist so programmiert, daß er 4 KByte an Daten von der Platte abrufen, während eine Benutzeranforde-

rung normalerweise 512 oder 1024 Byte lang ist. Damit werden weitere Datenanforderungen an den gleichen Bereich vorweggenommen.

Schreiboperationen werden zunächst nur im Cache-Speicher ausgeführt, wodurch kleinere Schreibforderungen zu größeren zusammengefaßt auf die Platte geschrieben werden können.

④ Elevator-Seeking

Anforderungen werden in einer Warteschlange so umsortiert, daß sich bei ihrer Abarbeitung eine möglichst optimale Führung des Schreib/Lese-Kopfes ergibt.

⑤ Mehrfache Plattenkanäle

Es können bis zur vier Plattenkanäle implementiert sein. Damit kann der Durchsatz beim Lesen und Schreiben bedeutend erhöht werden.

Erste Erfahrungen bei der Installation eines PC-Netzes

In PC-Netzen unter dem NetWare-Betriebssystem werden Dateien zentral auf Fileservern abgespeichert, die entsprechend den Zugriffsrechten von den einzelnen Nutzern angefordert werden können. Somit ist es möglich, Software nur auf den Fileservern abzuspeichern. Damit wird die Speicherkapazität der Arbeitsplatzrechner entlastet. Die entsprechenden Programme brauchen in diesem Fall nur einmal installiert werden. Auf Daten kann durch mehrere Nutzer quasiparallel zugegriffen werden.

Für die Betreuung des Netzes ist ein Mitarbeiter zu benennen, der die Berechtigung hat, im Supervisorstatus zu arbeiten. Das Paßwort des Supervisors ist nur diesem Mitarbeiter und dem staatlichen Leiter bekannt.

Für den Supervisor-Nutzer sind vor allem die menügesteuerten Programme SYSCON, FILER und QUEUE zu verwenden. Mit ihnen wird die Dateioorganisation und die Zuordnung der Zugriffsrechte vorgenommen.

In einem untersuchten PC-Netz werden folgende Aufgaben abgearbeitet:

- Büroautomatisierung

Zimmerreservierungssystem HODIS-2 unter Nutzung von SCOM-LAN

Dieter Lenz, Volker Nawrotzki, Martin Holzhauer
Interhotel Stadt Berlin

– Softwareentwicklung.

Zur Zeit sind in diesem Netz ein Fileserver und vier Arbeitsstationen mit unterschiedlichen Betriebssystemen angeschlossen. Mit diesem kleinen Netz sollen erste Erfahrungen gesammelt werden. Bei der Arbeit hat es sich gezeigt, daß die Zugriffszeiten im Netzbetrieb schneller sind, als im Einzelnutzerbetrieb.

Mit der Installation waren bis zum ersten Probetrieb ein bis zwei Mitarbeiter, die vorher nichts über die Konfigurationsmöglichkeiten und die Bedienung von NetWare wußten etwa einen Monat lang beschäftigt. Für die Nutzer war lediglich eine kurze Einweisung über die An- und Abmeldung im Netz notwendig.

Bei der Installation des Netzes wurde zunächst auf zehn anzuschließende Arbeitsstationen orientiert. Da es sich bei diesem Netz um ein nach oben offenes System handelt, können ohne größere Umstände zusätzliche Stationen in das Netz eingebunden werden (bis zu 100 Arbeitsstationen). Effektiv sollte man ungefähr 30 Stationen anschließen.

Literatur

- /1/ Chip Spezial „Lokale Netze“
- /2/ NetWare Product Information. Firmenschrift der Novell, Inc. Orem, UT 1986
- /3/ LAN Operating System Report 1986. Firmenschrift der Novell, Inc. Provo, UT 1987
- /4/ Grundlagen der Advanced NetWare 2.1. Firmenschrift der Novell GmbH.
- /5/ LAN TIMES. Monatszeitschrift der Novell, Inc.
- /6/ IBM PC Network Program. Technical Reference. Firmenschrift der IBM, Corp.

Ein verteiltes Datenverarbeitungssystem ist eine Systemlösung, bei der sich Dateneinheiten, Programmkomponenten sowie Peripherien auf mehreren informationell verbundenen Rechnern befinden.

LAN-gekoppelte Systeme bieten einen ökonomisch vorteilhaften und flexiblen Ansatz für die Entwicklung verteilter Systemlösungen. Die kooperierenden Rechnerknoten werden heute durch Personalcomputer (PC) mit eigenem Hauptspeicher und eigener Peripherie bestimmt.

In diesem Beitrag soll eine Anwenderlösung vorgestellt werden, die nicht nur einen wesentlichen Effektivitätsgewinn beim Einsatz dieser Rechentechnik sichert, sondern auch neue, für verbundene Systeme typische Leistungsmerkmale aufweist.

Die Lösung HODIS-2 ist eine Netzwerk-Applikation. Sie wurde auf der Basis des lokalen Rechnernetzes SCOM-LAN erarbeitet. Übersichtsinformationen zum Rechnernetz mit SCOM-LAN sind in /1/ zu finden.

HODIS bedeutet Hotelzimmer-Reservierungssystem. Es wurde zunächst als Insellösung für den Personalcomputer PC 1715 geschaffen /3/. Die computergestützte Rationalisierung des Verwaltungsprozesses Zimmerreservierung ist eine klassische Problemstellung für verteilte Systeme, wenn mehrere Systembenutzer zugelassen werden sollen. In großen Hotels und in zentralen Reservierungsbüros ist dies eine unverzichtbare Forderung. Typisch ist dabei der Vielfachzugriff von verschiedenen Endgeräten auf gemeinsam und koordiniert zu nutzende Datenbestände.

Der Inhalt der vorliegenden Veröffentlichung konzentriert sich auf den Systemübergang von der Einbenutzerlösung – auch als Sololösung bezeichnet – zur verteilten Netzwerk-Applikation.

Seit Januar 1988 befindet sich die LAN-Anwendung HODIS-2 zur Zimmerreservierung im territorialen Verkaufsbüro Berlin für die Interhotels Stadt Berlin, Berolina sowie Unter den Linden im Einsatz.

Zielstellungen für HODIS-2

Das Echtzeitsystem HODIS-1 (Sololösung) erlaubt die Reservierung von Hotelzimmern nach Zimmerarten im Dialog mit dem Disponenten. Es ist sowohl in einem Hotel als auch in Verkaufsbüros einsetzbar. Die gerätetechnische Grundlage bildet ein 8-Bit-Personal- oder Bürocomputer (z. B. PC 1715, BC A 5120, K 8924).

HODIS-1 verwaltet die Hotelkapazitäten von maximal neun Hotels mit insgesamt 30 Zimmerarten für einen Zeitraum von 512 Tagen (etwa 16 Monate). Es sind bis zu 5120 aktuelle Reservierungen speicherbar.

Das Reservierungssystem beinhaltet folgende Funktionen, die als eigenständige Programme von einem Grundmenü aus aufgerufen werden:

- Reservierung, Veränderungen, Stornierung
- Bildschirm- und Druckausgabe der freien Hotelkapazität einschließlich der Kontingente
- Bildschirmrecherche (Suchen nach Namen von Gast/Besteller)
- Kontingente anlegen/löschen.
- Auswertungslisten: Anreiseliste, Belegungsplan, Belegung nach Ländern, Anreiseübersicht, Belegung nach Gaststus

– Dateiwartung (Tageswechsel, Sicherheitskopie, Reorganisation).

Grundsätzlich werden alle Daten auf Disketten gespeichert und durch Verwendung eines für jeden Bediener vereinbarten persönlichen Codes gegen unberechtigten Zugriff gesichert. Das System HODIS-1 aktualisiert die Dateien während der Bearbeitung sofort.

Die wichtigsten Bewegungsdaten werden in den zwei Dateien FRKA (freie Kapazität) und RES (Reservierungsdatei) geführt.

Für jeden Tag des Buchungszeitraumes ist in FRKA ein Datensatz angelegt, in dem die freie Kapazität pro Zimmerart (z. B. Einbett-, Zweibettzimmer, ...) für alle im Solosystem enthaltenen Hotels eingetragen ist. Der Satzaufbau wird entsprechend der Anzahl Hotels und der Zimmerstruktur generiert. In der Datei RES ist für jede der 5120 mögli-

chen aktuellen Reservierungen ein Satz angelegt, der folgende Informationen enthält:

- An- und Abreisedatum
- Anzahl und Art der reservierten Zimmer
- Hotel
- Zuordnungsdaten wie Name von Besteller und Gast, Herkunftsland, Zahlart usw.

• jeder Reservierung ist es also notwendig,

- an der entsprechenden Stelle in FRKA die freie um die gebuchte Kapazität zu verringern,
- einen noch nicht belegten Satz in RES aufzusuchen (dieser Vorgang wird durch die Verwendung einer Zugriffsdatei erheblich beschleunigt) und
- die eingegebenen Daten dort abzuspeichern.

Die Datei STAHOD enthält alle Stammdaten des Systems wie:

- Stammdaten der Hotels (Hotelname, Anschrift, Telefon- und Telex-Nr. usw.)
- Informationen, die gebraucht werden, um den Zugriff auf die freie Kapazität zu ermöglichen (Daten für Adressrechnungen, Bezeichnungen der Zimmerarten)

In einem gesonderten Satz sind Daten gespeichert, die während eines Buchungstages konstant bleiben, sich aber beim Tageswechsel ändern (z. B. erster und letzter Tag des aktuellen Buchungszeitraums).

Alle Dateien werden bei der Systemgenerierung als leere Datentabellen (mit Leersätzen) angelegt.

Der Zugriff auf alle Dateien erfolgt in Form direkt adressierter Sätze.

Die gesammelten Erfahrungen beim Einsatz von HODIS-1 in gegenwärtig 13 Hotels und ähnlichen Einrichtungen in der gesamten DDR bestätigten die Tragfähigkeit dieses Dateikonzepts. Es erlaubt sogar, Forderungen der Anwender nach betriebsspezifischen Auswertungen der gespeicherten Daten durch das Einfügen zusätzlicher Funktionen in das System zu erfüllen. Wegen des stark strukturierten Programmaufbaus ist dies auch mit vertretbarem Aufwand zu erreichen.

Durch die Zentralisierung der Verkaufs-

aktivitäten in *Territorialen Verkaufsbüros* ergab sich die Notwendigkeit, auf der Grundlage der verfügbaren Gerätetechnik (8-Bit-BC/PC) ein Mehrarbeitsplatz-Reservierungssystem mit höherer Speicherkapazität kurzfristig zu schaffen. Wegen der Terminstellung von zwei Monaten kam eine Neuprogrammierung eines solchen Systems nicht in Frage. Die einzige Realisierungsmöglichkeit war, die bestehende umfangreiche Software des Einplatzsystems in das neu zu schaffende Mehrplatzsystem als Hauptkomponente zu übernehmen.

Bei der Planung des weiteren Vorgehens waren folgende Fragen zu klären:

1. Sind die Programmfunktionen *Datenverwaltung, Bearbeitung* und *Dialog* trennbar, um eine geteilte Verarbeitung auf getrennten Rechnern realisieren zu können?

2. Läßt das Dateikonzept einen Zugriff auf einen zentralisierten Datenbestand von mehreren Terminals zu, und sind Möglichkeiten für notwendige zusätzliche Informationen bezüglich einer Zugriffskennzeichnung vorhanden?

3. Können die Datenbestände im Gesamtsystem ohne umfangreiche Programmänderungen erweitert werden?

4. Welche technische Lösung zur Kopplung der Rechner ist die am besten geeignete? Zur Auswahl standen sternförmige Interface-Kopplungen über IFSS bzw. V.24 (Eigenentwicklung des Hotels Stadt Berlin) und das lokale Netz SCOM-LAN der Ingenieurhochschule für Seefahrt Warnemünde/Wustrow. Für die sternförmige Verbindung sprach, daß sie sich im Hotel schon für andere Rechnerkopplungen (z. B. UR-SADAT 5000 - PC 1715) bewährt hatte und bis ins Detail beherrscht wurde. Sollen als Dateiserver jedoch PC/BC verwendet werden, erweist sich als Nachteil, daß im Server je Terminal ein Interface-Anschluß erforderlich ist. Die Zuordnung der Terminals zu den Dateiservern kann ausschließlich hardwareseitig verändert werden. Die typische Übertragungstechnologie lokaler Netze vermeidet diese Nachteile durch einen gleichberechtigten Systemzusammenschluß von Servern und Terminals. Insbesondere ist der Hardwareaufwand der

Serverseite unabhängig von der Anzahl weiterer Netzwerkteilnehmer. Die mögliche flexible softwareorientierte Zuordnung von Terminal und Server ist als weiterer Vorteil zu nennen.

Um diese Fragen eindeutig beantworten zu können, wurden bei Teilfunktionen von HODIS-1, von denen zu erwarten war, daß ihre *Fernverarbeitung* besonders kritisch sein würde, die Befehle für den Diskettenzugriff durch eine provisorische DFÜ-Schnittstelle ersetzt und damit das Zeitverhalten beider Varianten der Kopplung im Vergleich zur Sololösung getestet.

Das Konzept der verteilten Verarbeitung der Bewegungsdaten auf mehreren Dateiservern, mit Zugriffsmöglichkeit für alle Terminals erwies sich als praktikabel und mit dem Dateikonzept von HODIS-1 vereinbar. Aufgrund der größeren Flexibilität und der Offenheit für spätere Erweiterungen versprach das SCOM-LAN bedeutende Vorteile gegenüber einer Standardinterface-Kopplung mit physischer Sternstruktur.

Das Softwarezentrum vom VEB Robotron Büromaschinenwerk Sömmerda vertreibt die SCOM-LAN-NIU (Network Interface Unit): 1715-NIU für die Personalcomputer PC 1715 und PC 1715W mit der Systemsoftware SCOM-NIOS (Network Interface Operating System) und einigen universellen Netzwerk-Applikationen.

Für Rechner mit K-1520-BUS (BC A 5120, K 8924 ...) liefert das Datenverarbeitungszentrum Rostock (VEB DVZ Rostock, Entwicklungs- und Produktionsstätte mikroelektronischer Rationalisierungsmittel) die NANOS-NIU.

Systementwurf für HODIS-2

Typisch für verteilte Netzwerk-Applikationen ist die Führung der Datenbestände in im Netz verteilten (passiven) Dateiservern und die Verarbeitung und Auswertung der Daten in (aktiven) Terminals.

Das von einem System verwaltete Datenvolumen kann durch das Hinzufügen weiterer Server (bzw. bei geringen Echtzeitforderungen die Nutzung mehrerer oder größerer Massenspeicher eines

Servers) vergrößert werden, ohne Programmänderungen vornehmen zu müssen.

Die sinnvolle *Arbeitsteilung* zwischen Server und Terminal beeinflusst entscheidend das Echtzeitverhalten des Systems. Daher sollte ihrer Gestaltung eine gründliche Analyse der Datenflüsse vorausgehen. Es ist dabei ein Optimum zwischen zwei gegensätzlichen Varianten zu finden: Ein intelligenter Server (Variante 1) entlastet das Netz (insbesondere beim Suchen und Sortieren in Dateien) von unnötigem Datentransfer – er steht aber andererseits anderen Teilnehmern für relativ lange Zeit nicht zur Verfügung, während er die komplexen, an ihn gerichteten Aufträge erledigt. Umgekehrt wird ein auf einfachste Grundfunktionen reduzierter Server (Variante 2) schnell wieder für andere Terminals frei, muß aber u. U. viel mehr Daten übertragen, die sich bei der Prüfung im Terminal als nutzlos erweisen (hohe Ringbelastung). Als Vorteil bei dieser Variante ergibt sich, daß ein komplexer Vorgang (z. B. ein Dateielement suchen) in viele Einzelaktionen (Verbindung aufbauen, Daten übertragen, Verbindung schließen) zwischen Terminal und Server zerlegt wird. Während das Terminal die empfangenen Daten auswertet, kann der Server quasi zeitgleich andere Terminals bedienen.

Die Zeit, die im SCOM-LAN für Verbindungsaufbau und Datentransfer RAM (Server) – RAM (Terminal) benötigt wird, ist gegenüber der Zeit für die Diskettenarbeit vernachlässigbar. Ein empfehlenswerter Weg ist es also, einen größeren Datenblock zunächst von der Diskette in den RAM des Servers zu lesen und danach portionsweise an das Terminal zu übertragen, wobei die erforderlichen Zeiger in den Terminals verwaltet werden müssen.

Es ist zu unterscheiden zwischen der Bearbeitung von Stamm- und Bewegungsdaten. Erstere werden beim Systemstart einmal als komplette Datei STAHO von den Terminals abgefordert und dort auf Diskette gespeichert.

Der häufige (Nur-Lese-)Zugriff der Terminals auf die Stammdaten belastet das Netz also nicht. Auf die Bewegungsdaten (Dateien FRKA und RES) dagegen muß satzweise zugegriffen werden, um die gleichzeitige Bearbeitung verschiedener Sätze derselben Datei durch mehrere Terminals zu ermöglichen.

Aus dem Gesagten ergibt sich, daß die Dateistruktur der Sololösung vollkommen erhalten bleiben konnte.

Es war zu prüfen, ob der gleichzeitige Zugriff auf denselben Satz einer Datei

durch verschiedene Terminals zu logischen Fehlern führen kann (bei der Reservierung ist z. B. der Doppelverkauf von Zimmern unbedingt zu vermeiden). In diesem Fall muß die Möglichkeit vorgesehen werden, einzelne Sätze bestimmter Dateien zeitweise zu sperren. Um dabei die gegenseitige Behinderung der Terminals gering zu halten, ist diese Sperrung auf das notwendige Maß an Umfang und Zeit zu beschränken. Die Kapazitätsdatei wurde solch eine Sperrfunktion geschaffen. Vom Termi-

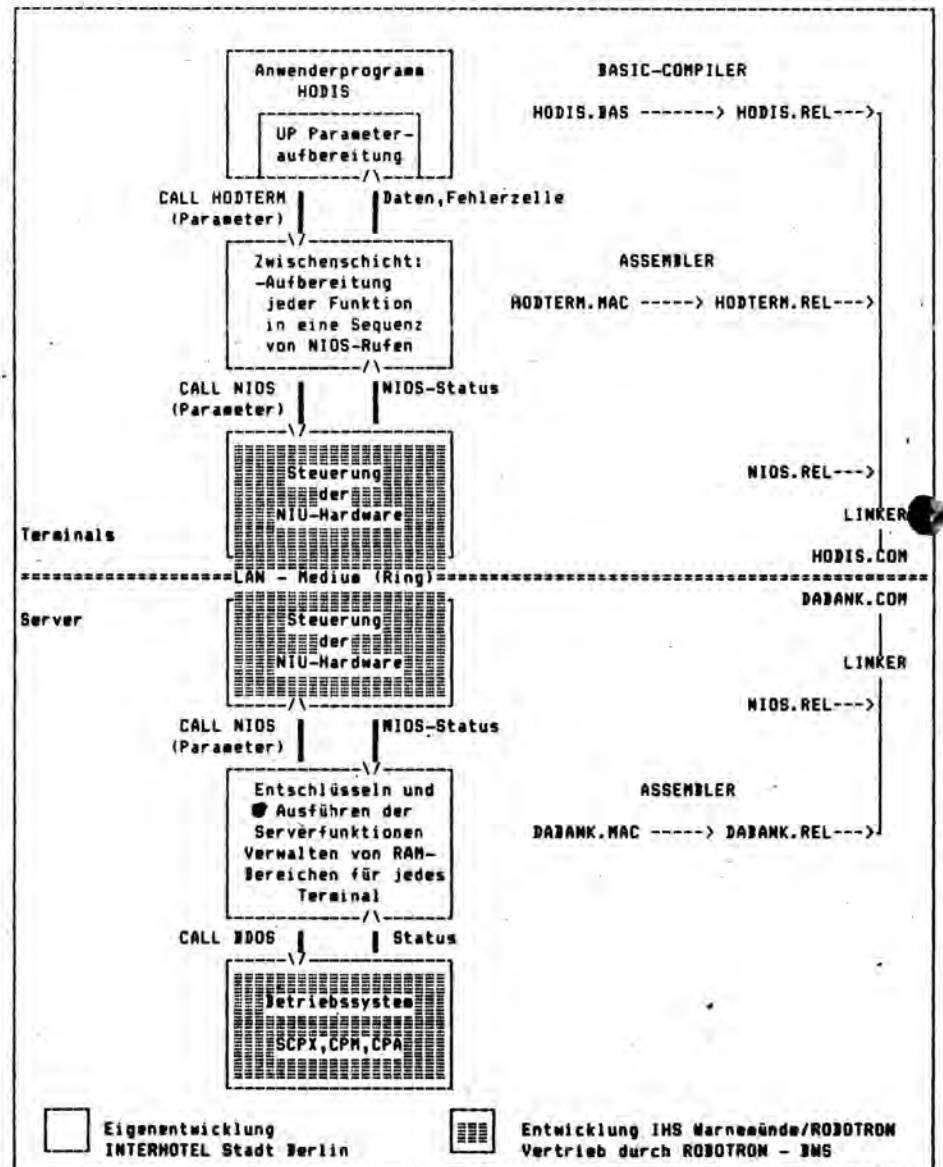


Abb. 1 Softwarekomponenten von HODIS-2-LAN

nal muß vor einer beabsichtigten Veränderung der freien Kapazität diese Sperrfunktion aufgerufen werden. Es ist unumgänglich, daß diese Sperrung bestehen bleibt, während der Disponent im Gespräch mit dem Gast die Angaben zur Reservierung im Dialog über die Tastatur erfaßt. Die Dauer der Sperrung eines Satzes ist daher im wesentlichen von der Konzeption des Terminaldialog abhängig. Blockierungen durch andere Terminals wirken sich – auch bei der Parallelarbeit von sechs Terminals – in der Praxis kaum störend aus. Eine entsprechende Meldung weist auf die Blockierung des Zeitraumes durch ein anderes Terminal hin. Die aus diesen Überlegungen resultierende Programmstruktur von HODIS-2 ist aus Abb. 1 ersichtlich.

Datei-Server

Da in der HODIS-2-Lösung bisher nur der Einsatz von vorhandenen 8-Bit-PC/BC mit Diskettenlaufwerken für den Server möglich war, mußte bei der Konzipierung der arbeitsteiligen Funktionen der Server so weit wie möglich entlastet werden. Nur dadurch ist unter Verwendung relativ langsamer Diskettenlaufwerke ein Mehrfachzugriff von mehreren Terminals gleichzeitig realisierbar.

Es wurden einfache Funktionen ohne weitere Bearbeitung im Server (Variante 2) gewählt:

- Anzahl von logischen Sätzen unterschiedlicher Dateien in entsprechende, dem Terminal zugeordnete Puffer lesen/bzw. aus den Puffern auf Diskette schreiben.

- Datenübertragung aus dem Dateipuffer des Servers zum Terminal mit einem Offset und minimal benötigter Länge (jeweils ein Datensatz).

Mit der Möglichkeit der Auswahl flexibler Datenmengen unabhängig von der konkreten Satzgröße einer Datei, wird nur die wirklich erforderliche Datenmenge für die Bearbeitung im Terminal übertragen. Dadurch ließen sich die Systemreaktionszeiten und die Netzbelastung optimal gestalten.

Für die Ermittlung von Reservierungsnummern übernimmt der Server (Va-

riante 1) durch einen Funktionsaufruf folgende Aufgaben:

- Suchen freie Reservierungszelle in einer Zugriffsdatei,
- Eintragen eines vom Terminal übergebenen Wertes (zwei Byte) an diese Position
- übertragen dieser Position (entspricht der Reservierungsnummer) an das Terminal.

Auf Grund der kurzen Bearbeitungszeiten ist eine quasiparallele Bearbeitung mehrerer Serveranfragen möglich. Sie wird durch die Reihenfolge des Eintreffens und ihrer Priorität geregelt.

Im Server befinden sich für sechs Terminals entsprechende eigene Dateipuffer und Sperrzellen für eine Datei.

Um einen flexiblen Zugriff von mehr als sechs Terminals zu gewährleisten, werden diese Puffer dynamisch verwaltet, d. h. ein Terminal belegt einen freien Puffer für eine technologisch abgeschlossene Handlung und gibt den Puffer durch Abmeldung frei.

Dadurch kann HODIS-2 in einem größeren Verbund betrieben und der Informationszugriff weiterer Terminals (Terminal im Hotelpfand, Terminals Büroautomatisierung für Leitungsinformationen u. s. w.) ermöglicht werden. Für diesen erweiterten Anwendungsfall sind im Server zusätzlich Überwachungsfunktionen des LAN implementiert, die im Falle einer Netzwerkstörung die Separierung des Teilringes *Reservierungslösung* vornehmen.

Entsprechend dem Einsatzfall können auf einem Server mehrere Datendisketten (entsprechend der Anzahl der Laufwerke) für unterschiedliche Einheiten (Hotels, Territorien u. s. w.) eingerichtet werden. Eine Datei- und Laufwerkswahl ist im Funktionstelegramm vorgelesen.

Es besteht auch die Möglichkeit, mehrere Server in einem System anzuordnen und die einzelnen Einheiten getrennt in verteilten Datenbeständen zu bearbeiten. Durch diese Verfahren läßt sich eine wesentliche Verbesserung des Echtzeitverhaltens in stark belasteten Projekten erzielen.

Terminal

Aus dem Einplatzsystem HODIS-1 wurden die Systembestandteile herausgelöst, die der Dateiwartung dienen und damit außerhalb des Echtzeitsystems abgearbeitet werden.

Alle weiteren Programmfunktionen sollten auf mehreren Terminals parallel abarbeitbar sein. Die Programme weisen bereits einen durchgehend strukturierten Aufbau auf, so daß eine Trennung zwischen Komponenten der Dialogführung und Abschnitten des Datenzugriffs einfach möglich war. Für das neue System HODIS-2 wurden die Dateizugriffe durch eine spezielle Parameteraufbereitung in Form von Unterprogrammen und Aufruf einer Zwischenschicht ersetzt. Aufgabe dieser Zwischenebene ist die Generierung und Weiterleitung von Serveranfragen unter Nutzung des Transportsystems SCOM-RIOS. Jede HODIS-Aktivität (z. B. Bereitstellen der freien Kapazität für einen Zeitraum) erfordert eine bestimmte Sequenz von Serveranfragen; jeder Funktionsaufruf zum Server erfordert einen Verbindungsaufbau. Einen Gesamtüberblick über diese Vorgänge im Vergleich zum einfachen Dateizugriff bei HODIS-1 gibt Tab. 1.

Die Zwischenschicht stellt also das Bindeglied zwischen den Anwenderprogrammen und der NIOS Transportschicht dar (vgl. Abb. 1).

Die Nutzoberfläche des Reservierungssystems wurde um die Möglichkeit der Auswahl der gewünschten Datendiskette und die Auswertung von Fehlern bezüglich der Datenübertragung erweitert.

Für jedes Terminal erfolgt im Rahmen der Generierung die Festlegung der Terminaladresse (Stationsnummer) und der von diesem Terminal erreichbaren Datendisketten (Stationsnummern Server und Diskettenlaufwerke) in einem Parameterfeld des Startprogramms. Alle weiteren Programme der Terminals sind identisch.

Dateiwartung

Die Datendisketten des Reservierungssystems HODIS-2 sind identisch mit de-

nen des Solosystems HODIS-1. Damit können die Funktionen der Dateiwartung

- Tageswechsel
 - Reorganisation
 - Sicherheitskopie der Datendiskette erstellen
- weiterhin mit Programmen von HODIS-1 außerhalb des Echtzeitsystems auf einem Terminal realisiert werden. Die Dateiwartung erfolgt für jede im System HODIS-2 enthaltene Datendiskette gesondert.

Systemeinsatz

Hardware

Die gerätetechnische Grundlage für das Reservierungssystem HODIS-2 bilden derzeit 8-Bit-PC/BC (PC 1715, BC A 5120 und K 8924).

Beim Einsatz als Terminal ist folgende Konfiguration erforderlich:

- 64 KByte RAM (Standard)
- 1 Diskettenlaufwerk MFS 1.6 (Standard)
- 1 Bildschirm 2K (Standard)
- 1 Standardtastatur (Es werden keine Funktionstasten verwendet)
- 1 Drucker K 6313, K 6314 oder EPSON LX-86
- 1 SCOM-LAN-NIU: 1715-NIU (Robotron BWS) / NANOS-NIU (DVZ Rostock).

Für den Einsatz als Server wird folgende Konfiguration benötigt:

- 64 KByte RAM (Standard)
- 2 oder 3 Diskettenlaufwerke MFS 1.6 (Standard)
- 1 Standardtastatur (nur für Programmstart)
- 1 Bildschirm
- 1 SCOM-LAN-NIU: 1715-NIU (Robotron BWS) / NANOS-NIU (DVZ Rostock).

Für das im Interhotel Stadt Berlin eingesetzte System HODIS-2 sind sechs PC 1715 als Terminals und zwei BC A 5120 als Server im Einsatz, wobei im Störfall ein Terminal (PC 1715) die Serverfunktion übernehmen kann.

Die Installation des LAN wurde unter Verwendung von handelsüblichen Materialien (AICu-Koaxialkabel, Einzel-

Arbeitsschritte Reservierung	HODIS-1	HODIS-2-LAN		
	Dateizugriffe in der Sololösung	Arbeitsteilung Terminal <--> Server	Serverzugriffe	FRKA Sätze gesperrt
Eingabe Anreise Zeitraum bzw. Abreise				
Bereitstellen freie Kapazität	Datei FRKA lesen	Sperren von Sätzen wenn gesperrt, Nachricht von welchem Terminal Datei FRKA in Serverpuffer lesen Teilsätze zum Terminal RAM T<--S RAM		
Eingabe Reservierungsdaten (Anzahl, Typ, Name Besteller, Gast ect.)	Eingabedialog (Zeitanteil ca. 90% der Gesamtreservierung)	Eingabedialog (Zeitanteil ca. 90% der Gesamtreservierung)		
Reservierung vornehmen	Suchen freie Reserv.Nr. in Zugriffsdatei Datei lesen, 2Byte eintragen, schreiben	Übertragen 2Byte T-->S Serv. sucht freie Res. Nr.; 2Byte eintragen T<--S Reserv.Nr.übertr.		
Aktualisierung freie Kapazität	Datei FRKA auf Diskette schreiben	Teilsätze FRKA z. Serv. RAM T-->S RAM Datei FRKA aus Puffer auf Disk. schreiben Entsperren der FRKA		
Reservierung abspeichern	Datei RES auf Diskette schreiben	Reserv. Satz T-->S Datei RES aus Puffer auf Disk schreiben		
Reserv. Beleg drucken				

koaxialstecker und Buchsen) problemlos realisiert.

In Vorbereitung späterer Erweiterungen wurde ein Teilnetz aufgebaut, das für den späteren Einbau von Überwachungs- bzw. Trennmöglichkeiten an einen zentralen Punkt geführt ist (Abb. 2).

Bedienung des Systems HODIS-2

Die tägliche Nutzung des Reservierungssystems erfolgt eigenständig durch die Mitarbeiter des Verkaufsbüros. Nach dem Starten der Server durch den Aufruf eines Startprogramms erfolgt eine automatische Überprüfung der eingelegten Datendisketten. Danach sind an den Servern keine weiteren Bedienhandlungen erforderlich.

Die Terminals werden anschließend

Abb. 2 Aufbau des LAN für HODIS-2

durch ein spezielles Startprogramm aktiviert, das folgende Aufgaben erfüllt:

- Ermittlung der im System enthaltenen Server und Anzeige der Arbeitsbereitschaft

- Übertragen der Stammdaten von den Servern zum Terminal und Abspeichern auf der im Terminal vorhandenen Programmdiskette

- Aktivieren des eigentlichen Reservierungssystems.

Die weitere Bedienung entspricht dem System HODIS-1.

Nach Beendigung der Arbeit an den Terminals durch Aufruf einer speziellen Programmfunktion werden die Server außer Betrieb genommen, wobei geprüft

wird, ob alle Terminals ihre Arbeit beendet haben.

Systemzuverlässigkeit und Datensicherung

Alle Daten des Systems werden auf Disketten geführt. Programmtechnisch ist gesichert, daß an den Server gesendete Daten sofort auf die Datendiskette geschrieben werden. Um Differenzen zwischen der freien Hotelkapazität und der Reservierungsdatei durch Geräteausfall zwischen den Schreibrufen für diese Dateien möglichst zu vermeiden, erfolgen diese Übertragungen zeitlich kurz hintereinander. Die kritische Zeit, während der ein Hardwareausfall zu Datenverlusten führen kann, ist damit sehr klein gegenüber der Gesamtbearbeitungszeit für einen Reservierungsvorgang.

Da weiterhin die verwendeten Dateien schon bei der Generierung des Systems mit allen Datensätzen (Leersätze) angelegt werden, ergeben sich während der Systembenutzung keine Änderungen der Dateikennsätze. Ein Datenverlust betrifft also maximal einen Datensatz einer Datei und damit genau einen konkreten Reservierungsvorgang.

Bei der täglichen Systembearbeitung stehen im Rahmen der Dateiwartung

Sicherheitskopien der aktuellen Datendisketten. Alle Reservierungsvorgänge werden an den Terminals zwangsweise protokolliert, was für die manuelle Beseitigung eventueller Havarien notwendig ist. Zur weiteren Erhöhung der Systemsicherheit wurden spezielle Prüf- und Korrekturprogramme erarbeitet.

Zeitverhalten

Das Echtzeitverhalten des Systems entspricht im wesentlichen dem der Sololösung HODIS-1. Für eine Reservierung benötigt ein ungeübter Disponent nicht mehr als etwa 90 bis 120 Sekunden, wobei Gastgespräch und Computerdialog etwa 90 bis 95 Prozent der gesamten Bearbeitungszeit beanspruchen.

Ein wesentlicher Vorteil von HODIS-2 ist die Möglichkeit der Abarbeitung von Rechercheprogrammen parallel zu den Reservierungsprozessen von anderen Terminals.

Die Anzahl der im LAN-Ring aktiven Terminals hat keinen merklichen Einfluß auf die am einzelnen Terminal zu erwartenden Antwortzeiten.

Aus dem jetzigen Betriebsverhalten der Systemlösung HODIS-2 kann eingeschätzt werden, daß ein weiterer Netzwerkausbau mit mehr als zehn Terminals ohne weiteres möglich ist.

Genügend Zeitreserven sind auch für eine zukünftige Einbeziehung anderer integrierter LAN-Applikationen zur Vervollkommenung von Verwaltungsprozessen vorhanden.

Zusammenfassung

Es hat sich gezeigt, daß der Aufbau einer verteilten Netzwerk-Applikation unter Verwendung vorhandener Programme unabhängig von der verwendeten Programmiersprache (BASIC, PASCAL ...) mit vertretbarem Aufwand in kurzer Zeit realisierbar ist.

Durch die Anwendung des Prinzips der verteilten Datenverarbeitung wurde eine erhebliche Steigerung der Leistungsfähigkeit des neuen Reservierungssystems gegenüber HODIS-1 erreicht. Bereits bei Einsatz von 8-Bit-Rechnern als Netzwerkserver-Stationen konnte das von einem System verwaltete Datenvolumen vervielfacht werden. Die Antwortzeiten im Vielfachzugriff auf die gemeinsam genutzten Datenobjekte sind durch eine anwendungsgerechte Aufteilung von Server- und Terminalprozessen im realen Betrieb vernachlässigbar gegenüber der Dialogführung. Zu diesem Ergebnis trug auch die Minimierung der Zugriffssperren auf Abschnitte der Serverdateien bei.

Ein großer Vorteil wird in der Einsatzmöglichkeit leistungsfähigerer Rechner (16-Bit-Rechner mit Hard-Disk) für die Serverkomponente gesehen, wobei die SCOM-LAN-NIU PROLAN für den EC 1834 als intelligente Netzwerkeinheit (U880A, 64...256 KByte) weitere Möglichkeiten für eine hardwaregestützte Parallelarbeit bietet.

Der Aufbau des Systems HODIS-2 war ohne detaillierte Kenntnisse der im SCOM-Transportsystems intern ablaufenden Prozesse durch die Nutzung der offenen Softwareschnittstelle SCOM-NIOS problemlos möglich.

Das in dieser Lösung verwirklichte Konzept der Server-Funktionen ist nicht nur auf den Anwendungsfall *Zimmerreservierung* anwendbar, sondern

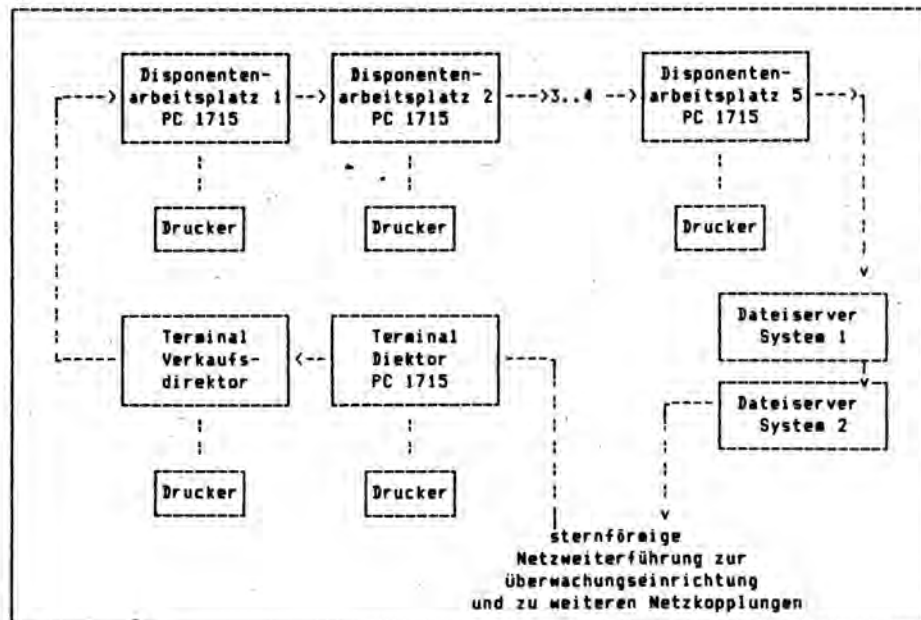


Abb. 3 Vergleich Dateizugriff HODIS-1/HODIS-2

Der Einsatz von lokalen Netzen in Betrieben

Axel Wüstemann, Wilhelm-Pieck-Universität Rostock
Institut für sozialistische Wirtschaftsführung, Informatik-Labor

bildet die Basis für weitere ähnlich gelagerte Anwendungen, z. B. in der Materialverwaltung.

- /1/ Göde, B.: SCOM-LAN – Ein lokales Rechnernetz der low-cost-Kategorie
- /2/ Göde, B.: Dokumentation der Systemsoftware SCOM-NIOS. SCOM-LAN-Handbuch; IHS Warnemünde/Wustrow, Direktorat WGB 1987
- /3/ Nawrotzki, V.; Holzhauer, M.: Hotelzimmer-Reservierungssystem HODIS-1. Betriebsdokumentation; Interhotel „Stadt Berlin“
- /4/ Löffler, H.: Lokale Netze. Akademieverlag, Berlin 1987;
- /5/ Löffler, H.: Rechnerverbundsysteme. Akademieverlag, Berlin 1984;
- /6/ Day, J. D.; Zimmermann, H.: The OSI Reference Model. Proceedings of the IEEE 71/1983/12, New York

Bedingt durch die geringe Verfügbarkeit von LAN-Hardware in der DDR konnten bisher in den Betrieben nur wenige Erfahrungen über den Einsatz lokaler Netze /1/ gesammelt werden. Existierende Rechnerverbundsysteme basieren meist auf der klassischen DFÜ-Technik /2/.

In diesem Beitrag wird versucht, die international verfügbaren Konzepte lokaler Netze hinsichtlich ihrer Einsatzmöglichkeiten in durchgängigen, betrieblichen Rechnerkommunikationssystemen zu klassifizieren. Durchgängige betriebliche Rechnerkommunikationssysteme werden in der Zukunft auch im Zusammenhang mit der rechnerintegrierten Fertigung (Computerintegrated Manufacturing, CIM) zunehmend an Bedeutung gewinnen. Charakteristisch für die Organisation eines CIM-Betriebes sind die Prozeßbezogenheit und der durchgehende Informationsfluß, durch den alle betrieblichen Teilprozesse verbunden sind.

Die Kommunikationsinfrastruktur eines Betriebes stellt eine wesentliche informationstechnologische Voraussetzung für die Integration betrieblicher Abläufe dar. Das Ziel einer solchen Infrastruktur besteht darin, betriebsweite Konsistenz, Aktualität und Verfügbarkeit von Informationen zu jeder Zeit und an jedem gewünschten Ort zu gewährleisten.

Einsatzcharakteristik lokaler Netze in Betrieben

Günstig für die Gestaltung der Kommunikationsinfrastruktur eines Betriebes ist ein hierarchisch organisierter Verbund verschiedener, der jeweiligen Teilaufgabe optimal angepaßter Teilsysteme, der ein Abbild der betrieblichen Organisationsstruktur ist. Eine solche Struktur ist kostengünstiger, da billigere Teilsysteme überall dort eingesetzt werden können, wo es die Anwendung erlaubt. Der meiste Verkehr spielt sich innerhalb der Teilsysteme ab, wodurch eine Lastreduktion des Gesamtsystems eintritt. Fehler in Teilsystemen führen nicht zum Zusammenbruch des Gesamtsystems. Eine geringere Störanfälligkeit und bessere Bedingungen zur Wartung und Pflege des Gesamtsystems sind die

Folgen. Insgesamt liegen günstigere Bedingungen für eine hohe Sicherheit gegen unautorisierten Zugriff auf die Datenbestände vor.

Vorteilhaft ist eine Verbindung von Teilnetzen auch aus Sicht des Systemplaners. Durch die Installation von zunächst unabhängigen Teilnetzen, die später zu einem Gesamtsystem verbunden werden ist der Aufbau des gesamtbetrieblichen Datennetzes stufenweise möglich. Dadurch kann die Komplexität der zu lösenden Probleme reduziert werden. Die bei der Installation der Teilsysteme gesammelten Erfahrungen können laufend ausgewertet und in das Gesamtkonzept eingearbeitet werden. Nicht zuletzt ist eine solche Struktur flexibler gegenüber den sich schnell ändernden technologischen Gegebenheiten.

Auf den verschiedenen Ebenen und innerhalb der einzelnen Ebenen selbst existieren ihrem Umfang und ihrer Art nach differenzierte Kommunikationsbedürfnisse, die an nachfolgenden Kriterien meßbar sind:

- **Durchsatzanforderungen**
 - relativ langsamer Zugriff auf große Datenmengen
 - schneller Zugriff auf große Datenmengen
 - schneller Zugriff (in Echtzeit) auf relativ kleine Datenmengen
- **anschließbare Endgeräte**
 - nur Computer mit Standard-Betriebssystemen (im Bürobereich, Planung, Verwaltung, Konstruktion usw.)
 - verschiedene (spezialisierte) Computer und Prozeßrechner (im Produktionsbereich, Betriebsdatenerfassung usw.)
 - andere Bürogeräte (z. B. multifunktionale Büroarbeitsplätze, Fernkopierer usw.)
 - Sensoren und Maschinen (im Produktionsbereich, Fertigungszellen)
- **Einbindung in übergeordnete Kommunikationssysteme**
 - horizontale und/oder vertikale Kommunikation
 - Heterogenität der Teilnetze und Endgeräte
- **Umgebungsverhältnisse**
 - elektromagnetisch saubere oder ge-

♦ Veranstaltung

5. Symposium „Grundlagen und Anwendung der Informatik“

Termin: 6. bis 8. Februar 1990

Ort: Karl-Marx-Stadt

Veranstalter: Technische Universität Karl-Marx-Stadt, Sektion Informatik

Schwerpunkte:

- Theoretische Grundlagen der KI
- Software-Entwicklungsumgebungen
- Compiler und Architekturen für prozedurale und logische Programmiersprachen (C, MODULA, LISP, PROLOG u. a.)
- Fragen der Wissensdarstellung, -verarbeitung und -gewinnung
- Kopplungen von KI-Sprachen mit klassischen Methoden (Datenbanken, Grafik, Geometrie u. a.)
- ausgewählte Anwendungen (Computer-Algebra, Expertensysteme, CAD/CAM, CIM).

Organisatorische Hinweise:

Weitere Informationen über Teilnahme und Vorträge erteilt die Technische Universität Karl-Marx-Stadt, Sektion Informatik, Tagungsleitung 5. Symposium, Postschloßfach 964, Karl-Marx-Stadt, 9010. Bei beabsichtigten Vorträgen wird bis zum 30. 6. 1989 eine kurze Inhaltsangabe benötigt.

störte Umgebung. Klimaverhältnisse usw. Die Umgebungsverhältnisse haben Auswirkungen auf das zu wählende Übertragungsmedium, wie z. B. Lichtwellenleiter bei starker elektromagnetischer Beeinflussung, auf die Gestaltung der Übertragungsprotokolle und eng damit verbunden auf die nutzbare Übertragungsrate.

- *Zuverlässigkeitsanforderungen und Anforderungen an den Datenschutz*
- *geographische Ausdehnung des Teilsystems.*

Auf der Grundlage der Gliederung eines (CIM-)Betriebes sollen fünf Arten von betrieblichen Teil-Kommunikationssystemen unterschieden werden:

- Netze zur Verbindung von Teilnetzen und zur Realisierung vertikal strukturierter Anwendungen (Rückgrat-Netze)
- Netze zur Realisierung eines schnellen Austausches großer Datenmengen im CAD- und F/E-Bereich
- Netze für die Büroautomatisierung
- Netze für die Produktionssteuerung
- Netze für die dezentrale Datenerfassung und -verteilung.

Für jede dieser Gruppen wird im weiteren eine Charakterisierung entsprechend den oben angegebenen Kriterien angegeben.

Rückgrat-Netze (Backbone-Netze)

Rückgrat-Netze /3/ bilden das Fundament der betrieblichen Rechnerkommunikation. Durch sie werden wesentliche Voraussetzungen für die Durchgängigkeit des betrieblichen Informationssystems realisiert. Das Rückgrat-Netz verbindet die Teilnetze der einzelnen Bereiche zu einem betriebsweiten Internetzwerk. Im Gegensatz zur direkten Kopplung der Teilnetze untereinander fällt durch sie das Problem der Wegsteuerung in einem allgemein vermaschten Netz weg. Die einzelnen Teilnetze werden als Endsysteme eines einfach strukturierten LAN angesehen. Das Rückgrat-Netz ist gleichzeitig die Basis für vertikal strukturierte Anwendungen.

Bedingt durch die große Anzahl angeschlossener Endsysteme und die Übertragung von zum Teil großen Datenmengen ist eine hohe Übertragungsrate erforderlich. Typische Werte liegen ab

zehn MBit/s. Breitbandnetze bieten mehr Durchsatz durch die Verwendung mehrerer Übertragungskanäle auf einer Leitung.

Es müssen die verschiedensten Endsysteme und Geräte anschließbar sein – Teilnetze, Großrechner, Terminals, Arbeitsplatzrechner, Spezialrechner wie Server und Gateway-Rechner usw. Über das Rückgrat-Netz ist eine Kopplung von LAN und digitaler Nebenstellenanlage (PBX) möglich. Es vermittelt den Übergang zu Weitverkehrsnetzen bzw. überregionalen LAN-LAN-Kopplungen und zum ISDN-Netz. Das Netz selbst kann so ausgelegt sein, daß eine Dienstintegration möglich ist.

Die Planung eines so komplexen Systems ist strategisch anzulegen. Von grundlegender Bedeutung ist dabei die Auswahl einer langlebigen, zukunftssicheren Technologie. Eine offene Architektur, basierend auf OSI-Standards, ist daher prinzipiell wünschenswert.

Man muß sich darüber im klaren sein, daß eine den gesamten Betrieb erfassende Rechnerkommunikation nichts Statisches ist. Jedes weitere Projekt in der Fertigungs- oder Büroautomatisierung erhöht den Vernetzungsbedarf. In jedem Falle wird das Rückgrat-Netz höhere Kommunikationsleistungen erbringen müssen.

Die Umgebungsverhältnisse sind sehr verschieden. Das Netz muß sowohl im Bürobereich, als auch im Produktionsbereich arbeiten können.

Rückgrat-Netze müssen hochgradig zuverlässig sein. Der zeitweilige Ausfall des Netzes kann große Teile der betrieblichen Rechnerkommunikation beeinflussen bzw. diese ganz unmöglich machen. Problemen des Datenschutzes ist größte Aufmerksamkeit zu schenken, da jeder Nutzer potentiell auf beliebige Datenbestände zugreifen kann.

Die geographische Ausdehnung kann sehr weitläufig sein, da alle betrieblichen Bereiche erfaßt werden müssen.

Netze für den CAD- und F/E-Bereich

Im CAD- und F/E-Bereich sind große Datenmengen, wie Geometriedaten, Normteil- und Stücklisten-Dateien, die im Bereich bis zu einigen MByte liegen,

zu übertragen. Dem Ingenieur müssen diese Daten während des Entwurfsprozesses, sozusagen in Echtzeit, bereitgestellt werden. Die angeschlossenen Endgeräte haben oft eine hohe Verarbeitungsleistung, der die Übertragungsrate des sie verbindenden Netzes angepaßt sein muß. Das Netz muß daher einen schnellen Zugriff auf große Datenbestände zulassen. Übliche Übertragungsraten liegen zwischen fünf und einigen zehn MBit/s.

Die angeschlossenen Endgeräte sind relativ homogen, meist unter Steuerung eines Standardbetriebssystems.

Die Netze für CAD- und den F/E-Bereich können als relativ geschlossene Systeme aufgefaßt werden. Der meiste Verkehr findet innerhalb des Netzes statt, nur gelegentlich ist der Zugriff auf Datenbestände z. B. eines Großrechners erforderlich. Der Einsatz OSI-gerechter Protokolle ist daher nicht immer notwendig, aufgrund der geforderten Übertragungsleistung auch nicht immer sinnvoll. Vielmehr sollte das gesamte Netz nach außen als OSI-gerechtes Endsystem erscheinen.

Die Umgebungsverhältnisse sind grundsätzlich sauber.

Natürlich muß die Zuverlässigkeit eines solchen Netzes hoch sein. Ein zeitweiliger Ausfall hat aber nur Auswirkungen innerhalb des Bereiches, in dem das Netz installiert ist. Maßnahmen zum Datenschutz sind kaum notwendig, es sei denn, verschiedene Projektgruppen müssen voneinander abgeschirmt werden.

Die geographische Ausdehnung ist meist gering, auf eine Abteilung oder Projektgruppe beschränkt.

Netze für die Bürokommunikation

Im Bürobereich sind mittlere bis große Datenmengen im Bereich einiger KByte bis einiger 100 KByte, vor allem in Form von Texten (Dokumente) und Datenbankrelationen zu übertragen. An die Übertragungsrate werden aber nicht so hohe Anforderungen gestellt. Mögliche Werte liegen zwischen einigen 100 KBit/s bis zu zehn MBit/s. Die Anforderungen werden aber mit neuen Anwendungen, wie z. B. Desktop Publis-

hing, in der Zukunft steigen. Die Grenze zwischen CAD- und Büronetzen ist fließend.

Es sind neben den LAN auch PBX-Anlagen einsetzbar. Sie bieten zusätzlich die Möglichkeit der Dienstintegration, so daß auch multifunktionale Arbeitsplätze eingesetzt werden können. Ansonsten sind typische Endgeräte Personalcomputer mit Standardbetriebssystemen, wie CP/M und MS-DOS sowie Dateiserver.

Für die weitere Charakteristik gilt dasselbe wie für die Netze im CAD- und F/E-Bereich. Lediglich dem Datenschutz kommt oft eine größere Bedeutung zu, da nicht jedem Sachbearbeiter jede Information zugänglich sein sollte. Oft sind im Büro aber auch Billig-Systeme einsetzbar, insbesondere dann, wenn es lediglich darum geht, teure Peripheriegeräte für mehrere PC nutzbar zu machen oder bestimmte Daten auf Hintergrundspeicher zu archivieren, um sie dann zur Verarbeitung auf einen PC zu kopieren.

Netze zur Produktionssteuerung

Zur Maschinen- und Robotersteuerung und zur Übertragung von Prozeßinformationen ist es notwendig, kleine Datenmengen im Bereich einiger Byte bis kByte auch in Echtzeit zu übermitteln.

Die angeschlossenen Endsysteme sind sehr heterogen: PC, Prozeßrechner mit Multitask-Betriebssystemen zur Zellensteuerung, speicherprogrammierbare Steuerungen, Maschinen, Roboter und Sensoren. Standardisierte Schnittstellen und Kommunikationsverfahren sind also sehr wichtig.

Die Einsatzumgebung der Netze ist u. U. stark gestört.

Fertigungsnetze sind geschlossene Teilsysteme, z. B. für eine Fertigungszelle. Es finden jedoch relativ häufig Interaktionen mit übergeordneten Systemen der Produktionssteuerung (z. B. Leitstand) statt.

Die Zuverlässigkeit muß besonders dann sehr hoch sein, wenn der durch den Ausfall des Netzes bedingte Fertigungsausfall Auswirkungen auf weitere Bereiche der Fertigung hat. Maßnahmen zum Datenschutz sind nicht notwendig.

Die geographische Ausdehnung von Fertigungsnetzen ist oft sehr gering, z. B. auf eine Fertigungszelle begrenzt.

Netze zur dezentralen Datenerfassung und Datenverteilung

Zur Betriebsdatenerfassung (BDE) werden an verschiedenen Stellen im Fertigungsbereich spezielle Betriebsdatenterminals (BDT) aufgestellt. Bei den erfaßten Daten handelt es sich hauptsächlich um Daten, die mit dem Produktionsfortschritt und der Maschinenbelegung zusammenhängen. Je nach Anzahl der BDT können die dabei entstehenden Datenmengen recht groß sein. Besondere Anforderungen an die Übertragungsgeschwindigkeit sind dabei jedoch nicht zu stellen.

Die angeschlossenen Endgeräte sind meist homogen. Das gesamte System *Leitrechner, BDT und Netz* wird oft von einem Hersteller bezogen. Typischerweise sind die BDT über ein Liniennetz an den Leitreechner angeschlossen, der die gesamte Steuerung des Netzes übernimmt (Abfrage der BDT nacheinander), so daß spezielle LAN-typische Zugriffsverfahren nicht notwendig sind. Der Leitreechner ist als normales Endgerät in ein LAN integriert. So gesehen können die BDT als Peripherie des Leitrechners aufgefaßt werden.

Die Einsatzumgebung kann stark gestört sein. Echtzeitanforderungen bestehen nicht.

Die Zuverlässigkeit des BDE-Systems muß sehr hoch sein, da ein Ausfall große Auswirkungen auf die weitere Fertigung haben kann. Maßnahmen zum Datenschutz sind nicht erforderlich.

Die geographische Ausdehnung kann gering gehalten werden, wenn z. B. pro Fertigungsbereich ein BDE-System installiert wird.

Typologie lokaler Rechnernetze und ihre Eignung für betriebliche Bereiche

Gebräuchliche Merkmalskombinationen

In der Praxis treten typische Kombinationen für die Parameter der oben angegebenen Merkmale auf, nach denen sich

die meisten LAN-Produkte sinnvoll klassifizieren lassen.

① Abfrage; Sterntopologie; Zweidrahtleitung; geringe Übertragungsrate; keine Echtzeit; Basisband; billig

Im Durchschnitt bis zu sechs PC können über V.24-Leitungen an einen zentralen PC angeschlossen werden. Die Übertragungsrate liegt entsprechend bei 9,6 KBit/s (beim AT 56 KBit/s).

Übliche Anwendungen sind das Teilen von Geräten, wie Drucker, Plotter oder Hintergrundspeicher, für Datenarchivierung. Ein Beispiel ist das Produkt EasyLAN /5/.

② Abfrage-, CSMA/CD-, Token- oder andere Zugriffsverfahren; Bus- oder Ringtopologie; Zweidrahtleitung oder Koaxialkabel; geringe Übertragungsrate; keine Echtzeit; Basisband; billig

Das wichtigste Merkmal dieser Gruppe ist wie bei ①, daß diese Netze billig sind /6/. Ein Beispiel aus der DDR ist das SCOM-LAN. Der Anschluß eines PC 1715 an dieses Netz kostet etwa 400 M.

Entsprechend ihrer Einsatzcharakteristik erfüllen solche Netze oft vorliegende, einfache Einsatzanforderungen in der Büroarbeit, insbesondere in kleinen Betrieben. Sie sind auch zur Betriebsdatenerfassung und -verteilung einsetzbar.

③ CSMA/CD- oder Tokenverfahren; Bus- oder Ringtopologie; Zweidrahtleitung, meist Koaxialkabel; Übertragungsraten; Echtzeit möglich; Basisband; mittlere Kosten

Der untere Bereich dieser Gruppe von Netzen wird von solchen Systemen wie ROLANET1 /7/ oder LOTUNET /8/ gebildet. International wird diese Klasse durch Systeme wie StarLAN oder den IBN-Token-Ring /9/ vertreten.

Sie sind vorrangig für Büroanwendungen und für einfache CAD-Systeme auf der Basis von PC geeignet. Übliche Steuerungssoftware sind Netzbetriebssysteme der Leistungsklasse von SCPNet /7/ oder MicroNET-80.

④ CSMA/CD- oder Tokenverfahren; Bus- oder Ringtopologie; Koaxialkabel; hohe Datenübertragungsrate; Echtzeit möglich; Basisband; mittlere bis hohe Kosten

In diese Klasse gehört das Ethernet. Es ist das universellste LAN oder auch als einfaches Rückgrat-Netz nutzbar.

Eine sehr leistungsfähige Implementierung, das SK-NET, liefert die BRD-**Ma Schneider & Koch/11/**. Die Slot-Karte für den PC ist einem 32-Bit-Mikroprozessor MC 68000 und 128 kByte RAM sowie den VLSI-LAN-Controller (LANCE) AMD 7990 ausgerüstet. Die Steuerung des Netzes übernimmt das Netzbetriebssystem NetWare. Diese Karte kommt vorrangig in einer Serverstation zum Einsatz. Für die Arbeitsstationen sind die wesentlich billigeren SK-NETJunior-Karten ohne eigenen Prozessor besser geeignet. Es existieren Protokollübergänge zu UNIX-typischen Netzen mit TCP/IP-Protokollen /12/ und zum Network-File-System (NFS) /13/ (ein Netzbetriebssystem für UNIX-basierte Grafikworkstations von SUN) sowie zur DEC-Welt.

Das sich in Entwicklung befindende ROLANET2 von Robotron soll ebenfalls vollständig dem Ethernet-Standard entsprechen.

In diese Kategorie gehört auch das TOP /14/.

Ein leistungsfähigeres System dieser Klasse, speziell für den Einsatz im CAD-Bereich konzipiert, ist der DOMAIN-Token-Ring für APOLLO-Grafikworkstations /15/. Die Bruttoübertragungsrate von zwölf MBit/s wird auch bei hoher Last voll ausgenutzt. Das dazugehörige Netzbetriebssystem *Network Computing Architecture (NCA)* erlaubt unter anderem die dynamische Auslagerung von Rechenprozessen auf andere (gerade freie) Workstations (Prozeßmigration).

⑤ Tokenbusverfahren; Bustopologie; Koaxialkabel; mittlere Übertragungsrate; Echtzeitfähig; Trägerbandübertragung; mittlere bis hohe Kosten

Das sind die Merkmale des MiniMAP /19/. Wesentliche Unterschiede zum Breitband-MAP sind die reduzierte Da-

tenübertragungsrate auf 5 MBit/s und die nicht vollständig ausgebaute OSI-Protokollhierarchie. Es existieren nur die Schichten 1, 2 und 7. Dadurch wird eine kürzere Reaktionszeit erreicht. MiniMAP ist speziell als Teilnetz des MAP zur Zellensteuerung konzipiert.

⑥ Tokenbusverfahren; Baumtopologie; Koaxialkabel; hohe Übertragungsrate; nicht echtzeitfähig; Breitbandübertragung; hohe Kosten

Das Breitband-MAP /15/ ist seit der Version 3.0 hauptsächlich als Rückgrat-Netz gedacht. Die Version soll für die nächsten sechs Jahre eingefroren werden, um Hardware-Entwicklern die Chance zu geben, preisgünstige VLSI-Chips herzustellen. Die Akzeptanz von MAP wird wesentlich von den Kosten pro Knoten abhängen.

Ein Breitband-MAP Starterkit, bestehend aus zwei Interface-Karten einschließlich Software, MMS-Software und vorkonfektioniertem Kabel sowie der Kopfstation, wurde auf der Hannover-Messe Industrie 1988 für 47 014 DM angeboten /17/.

⑦ Tokenringverfahren; Ringtopologie; Lichtwellenleiter; sehr hohe Übertragungsrate; Echtzeit möglich; Trägerband; sehr hohe Kosten

Gegenwärtig werden in den USA Forschungsarbeiten zur Standardisierung eines Hochleistungsnetzes mit Lichtwellenleiter unternommen. Es trägt die Bezeichnung *Fiberoptic Distributed Data Interface (FDDI)*. Mit einer Datenübertragungsrate von 100 MBit/s wird FDDI für weit in die Zukunft reichende Anwendungen konzipiert. FDDI soll neben seiner Rolle als Rückgrat-Netz auch zur schnellen überregionalen LAN-LAN-Kopplung, den sogenannten *Metropolitan Area Networks (MAN)*, dienen. Die Notwendigkeit für eine Leistungsfähigkeit, wie sie FDDI bietet, wird vor allem im Zusammenhang mit Anwendungen in der Bildverarbeitung und Künstlichen Intelligenz gesehen. Man rechnet für Anfang der 90er Jahre mit einem ersten Bedarf in den USA. Es finden sich bereits erste VLSI-Hersteller, die an einem Chip-Satz für FDDI

arbeiten. Es wird eingeschätzt, daß Hersteller wie DEC und IBM die ersten sein werden, die FDDI-Netze einsetzen /18/.

Spezielle Systeme

Außer dem vorgestellten System gibt es eine Reihe Systeme, die für durchgängige betriebliche Rechnerkommunikationssysteme wichtig sind, sich aber nicht so einfach in das oben angegebene Schema einfügen.

• Anschluß von Großrechnern an LAN

Mit der an der TU Dresden entwickelten ESER-Lokalnetz-Koppeleinheit ELKE /19/ ist der Anschluß eines ESER-Rechners an ROLANET1 möglich.

Es besteht aus Baugruppen des K 1520-OEM-Systems. Der ESER-seitige Anschluß erfolgt über eine SIF-ESER-Schnittstelle, der LAN-seitige über den LNC1. Auf der ESER-Seite wird ein OSI-entsprechender Transportdienst bereitgestellt.

• BDE-Systeme

Das Betriebsdatensystem (BDS) Robotron A 5230 /20/ ist sowohl für die Datensammlung und -verteilung als auch für die Produktionssteuerung geeignet. Die Zentrale des Systems bildet eine Systemsteuereinheit, ausgerüstet mit einem 16-Bit-Prozessor U8000, insgesamt 1,6 MByte Arbeitsspeicher, maximal 160 MByte Massenspeicher sowie Standardperipherie. Bis zu 24 Bildschirmterminals können sternförmig und bis zu 60 BDT durch ein maximal drei km langes Linieninterface an einen mit der Zentrale verbundenen Konzentratorenrechner angeschlossen werden. Die Systemsteuereinheit läuft unter dem UNIX-kompatiblen Betriebssystem MUTOS. Über V.24 oder IFSS ist das BDS in übergeordnete Systeme integrierbar.

Vom VEB Forschung, Entwicklung und Rationalisierung (FER) des Schwermaschinen- und Anlagenbaus wird eine BDE-Lösung, basierend auf PC 1715 als Leitrechner und KC 85 als BDT angeboten. Die Software erlaubt die Ermittlung und grafische Darstellung des aktuellen

Bearbeitungsstandes auf einer Maschine sowie der Be- und Entlastung von Bearbeitungsstationen /21/.

• Systeme zur Produktionssteuerung

Auf der Basis des Microcontrollers 8044 von Intel wurde 1984 von dieser Firma unter der Bezeichnung *Bitbus* eine Hochleistungs-Kommunikationseinrichtung für verteilte Automatisierungssysteme definiert. Mit ihr lassen sich bis zu 250 Knoten miteinander verbinden. Im Synchronbetrieb lassen sich bis zu 30 m, im selbstgetakteten Betrieb bis zu 1000 m überbrücken. Es werden drei verschiedene Übertragungsraten zugelassen: 62,5, 375 und 2400 KBit/s. Als Übertragungsmedium werden verdrehte Leitungen oder auch zehnpolige Flachbandkabel eingesetzt. Durch entsprechende Treiber kann der Bitbus in verschiedene, vornehmlich Multitasking-Echtzeitbetriebssysteme eingebunden werden /22/.

Eine in der DDR entwickelte Lösung wird in /23/ vorgestellt.

Zusammenfassung

Nach Auffassung des Autors könnten bei genügender Verfügbarkeit mit den derzeit in der DDR existierenden LAN-Lösungen auf breiter Ebene Erfahrungen beim Einsatz lokaler Netze in Teilbereichen unserer Betriebe und Kombinate gesammelt werden. Diese Erfahrungen sind auch nicht durch sorgfältigstes Literaturstudium zu ersetzen.

Die Situation würde sich mit einer baldigen Verfügbarkeit von ROLANET2 weiter entscheidend verbessern. Damit ließen sich für viele Betriebe auch schon durchgängige Netze schaffen.

Die internationale Entwicklung läßt den Versuch erkennen, immer leistungsfähigere Systeme immer kostengünstiger zu realisieren (MAP) und zu neuen Leistungsdimensionen vorzudringen (FDDI).

Eine der Schwergewichte der internationalen Entwicklung liegt auf einer der wesentlichen Voraussetzungen für die Schaffung durchgängiger betrieblicher Rechnerkommunikationssysteme: der Integrationsfähigkeit von Teilsystemen.

MAP und TOP sind Versuche, dieses Problem auf der Basis von OSI zu lösen.

Auf der Basis der IEEE-LAN-Standards versuchen die Anbieter, ihre Systeme zu öffnen. Übergänge zwischen verschiedenen Industriestandards (vgl. SK-NET) bzw. die Einigung auf solche, bestimmen das Bild. Viele Hersteller streben Migrationsstrategien ihrer eigenen Kommunikationsarchitekturen zu OSI an (z. B. DEC und HP). Auf jeden Fall wird es in absehbarer Zukunft mehr Standards auf dem Gebiet der LAN geben als nur OSI.

Literatur

- /1/ Wüstemann, A.; Rossa, G.: Lokale Netze – Ein Überblick. rd 24(1987)9, S. 31; 10, S. 31
- /2/ Bolz, F. R.: Datenaustauschkonzept für einen heterogenen Rechnerverbund. VI. Kolloquium LO+EDV, Kurzfassungen, WPU Rostock 1988, S. 14–16
- /3/ Martini, P.: Das aktuelle Schlagwort: Backbone-Netze. Informatik-Spektrum (1987/10), 100–102
- /4/ Segl, E.: Zellenstruktur am MAP-Backbone. Elektronik 1986, Heft 10, S. 128–132
- /5/ Feldt, T.: \$100 software links IBM-PC to LAN. Electronics January 22, 1987, S. 101
- /6/ Barney, C.: Low speed LAN's check in at \$100 per connection. Electronics July 10, 1986, S. 29–30
- /7/ Richter, J.; Terpe, B.: ROLANET1 – Ein lokales Netz des VEB Kombinat Robotron. NTB 30(1986)5, S. 146–149
- /8/ Weidhase, F.; Klempien, R.: Hardwarekomponenten des lokalen Netzes LOTUNET. WZ des IZ des Hochschulwesens an der TU Dresden 1(1987)2, S. 5–15
- /9/ Token-Ring Network PC Adapter, Technical Reference IBM Personal Computer Hardware Reference Library. Firmenschrift der IBM, 1986
- /10/ IBM PC Network Program User's Guide. Firmenschrift der IBM, Corp.
- /11/ Top Performance im PC-Netzwerk, Ethernet-LAN, SK-NET und SK-NET junior. Firmenschrift der Schneider und Koch GmbH
- /12/ Holler, E.: Transportprotokolle im lokalen Netzbereich. informationstechnik it, 29(1987)6, S. 379–385
- /13/ Lippert, M.: NFS – Der Industriestandard für Netzwerk-File-Systeme. unix/mail 5(1987)1, S. 35–39
- /14/ Löffler, H.: MAP und TOP – Architekturkonzepte für lokale Datenkommunikation. rd 25(1987)7, S. 13–16
- /15/ Das Konzept, mit dem APOLLO DOMAIN Rechnerwelten verändert hat. Firmenschrift der APOLLO DOMAIN Computer GmbH
- /16/ Prager, M.; Schumacher, H.: Experimentalsystem für MiniMAP, unveröffentlichtes Manuskript, WPU Rostock

/17/ Fabrikautomation mit MAP – Informationen zur Hannover-Messe Industrie 1988

/18/ Warren, C.: Fiberoptic networks shine new light on high-speed transfers. Computer Design June 15, 1987, S. 45–52

/19/ Wulst, N.; Herold, A.: Aufbau und Arbeitsweise der ESER-Lokalnetz-Koppeleinheit ELKE. WZ des IZ des Hochschulwesens an der TU Dresden 1(1987)2, S. 79–80

/20/ Hoffmann, H. u. a.: Das Betriebsdatensysteme AS230 – Eine universelle Systemlösung zur Rationalisierung betrieblicher Leitungs- und Steuerungsprozesse. NTB 32(1988)1, S. 21–25

/21/ VEB FER, Zentrale Leitstelle für Information und Dokumentation, Nachnutzungsangebot 1988

/22/ Bodenkamp, J.: Bitbus: Verteilte Intelligenz in der Automatisierung. Elektronik 16, 10. 8. 1984, S. 115–120

/23/ Beikirch, H. u. a.: Intelligente dezentrale Prozeßkoppelmodule. Feigerätetechnik 36(1987)8, S. 360–362

♦ Veranstaltung

12. Wissenschaftlich-Technische Konferenz Automatisierungssysteme

Termin: 16. bis 17. 11. 1989

Ort: Magdeburg

Schwerpunkte:

- Automatisierungssysteme
- Kommunikationssysteme
- LAN und Feldbussysteme und Möglichkeiten ihrer Kombination
- neue gerätetechnische Lösungen für Automatisierungsanlagen auf der Basis von 16-Bit-Systemen in Kopplung mit technischen Prozessen
- ASIC – Anwendung in der Automatisierungstechnik

Organisatorische Hinweise: Anfragen sind zu richten an die KDT, WGMA, Clara-Zetkin-Str. 115/117, PSF 1315, Berlin, 1086

Drei Beiträge aus LO + EDV '88

Prof. Dr. Gerd Rossa, Prof. Dr. Martin Graef
Frank Bolz

MBASE – Ein Methodenbanksystem als Shell des DES

(Prof. Dr. Gerd Rossa,
Wilhelm-Pieck-Universität Rostock)

Systembeschreibung

Das System MBASE ist Bestandteil des DES-Komplexes und ein softwaretechnologisches Werkzeug zur Beherrschung der Integrationsprobleme, die bei dem Einsatz verschiedenster Applikationen über einer verteilten heterogenen Datenbasis entstehen. Das gilt insbesondere für Probleme, die bei der Verfolgung eines CIM-Konzeptes, das verteilte Verarbeitung voraussetzt, auftreten. Sie ergeben sich sowohl auf dem Niveau der Datenbereitstellung als auch bei der Kopplung von Programmen (Methoden). MBASE besteht im wesentlichen aus dem Datenbankkern DES und einem maskengesteuerten Methodenmonitor TINA. Die Verbindung zwischen Programm (Methode), Methodenbeschreibung und dem DES wird für den Nutzer sehr komfortabel über Masken realisiert (Abb. 1).

Zur Methodenauswahl bzw. Interpretation der Ergebnisse kann ein Schlußfolgerungsmechanismus eingesetzt werden. Diese DES-Komponente (LOGI) verarbeitet Regeln und Fakten in einer PROLOG-Notation (vgl. rd 4/89).

Die Systemarchitektur ist so angelegt, daß Funktionsumfang und Nutzerkomfort den integrierten Systemen nahekommen, jedoch mit dem Versuch, eine offene Architektur zu realisieren, d. h. der Nutzer kann auch neue Programme (Methoden) in das System integrieren. Durch diese offene Systemarchitektur und die Vielzahl der vom DES angebotenen Datentypen sind die Anwendungen des Gesamtsystems sehr vielfältig. Es lassen sich insbesondere solche Probleme bearbeiten, die durch den Einsatz von Datenbanksystemen charakterisiert sind. Außerdem lassen sich durch die Bereitstellung einer PASCAL-Schnittstelle zum DES beliebige Programme in den Methodenkomplex einbinden.

Entwicklungsziele von MBASE

Entscheidungs-Unterstützungs-Systeme (DSS) verlangen eine flexible Handhabung verschiedenster Methoden, Ver-

fahren, Programme und Modelle im Prozeß der Entscheidungsunterstützung.

Deshalb stellen die Methodenverwaltungssysteme eine wichtige Komponente der DSS dar.

Spezielle Forschungsziele, die durch die Untersuchungen am MBASE erreicht werden sollen, sind deshalb:

- Definition, Klassifizierung, Aufgabengebiete und Struktur von MBS in Entscheidungs-Unterstützungs-Systemen
- Kopplung von MBS mit DBS (z. B. mit DES)
- Methodenauswahl-Möglichkeiten
- XPS als Methodenmonitor (XPS = Expertensystem)
- Nutzung von LOGI zur Auswahl von
 - Methoden-Problemen
 - Ergebnisinterpretation
 - Datenauswahl
 - Methodenbeschreibung
 - Anwendung-Leitungsinformationssystem
- Entwurf eines theoretischen Modells kooperierender MBS auf LAN-Basis.

Struktur von MBASE

Die Verwaltung von Methoden wird auf der Basis des DES realisiert. Hierzu werden insbesondere der Datentyp Text (für Erklärungen) und der Datentyp Bulk (für die eigentlichen Programme) herangezogen.

Teilfunktionen einer Methodenverwaltung sind:

- Formulieren von Methoden; explizite Formulierung durch den Nutzer
- automatische Methodenkonstruktion
- Modifikation der Methoden durch den Nutzer
- Wartung der Methoden
- Verknüpfen von Methoden
- Aktivieren von Methoden; Aufruf und Parameterübergabe durch den Nutzer
- Aufruf und Parameterübergabe durch das System.

Zur Realisierung der Methodenverwal-

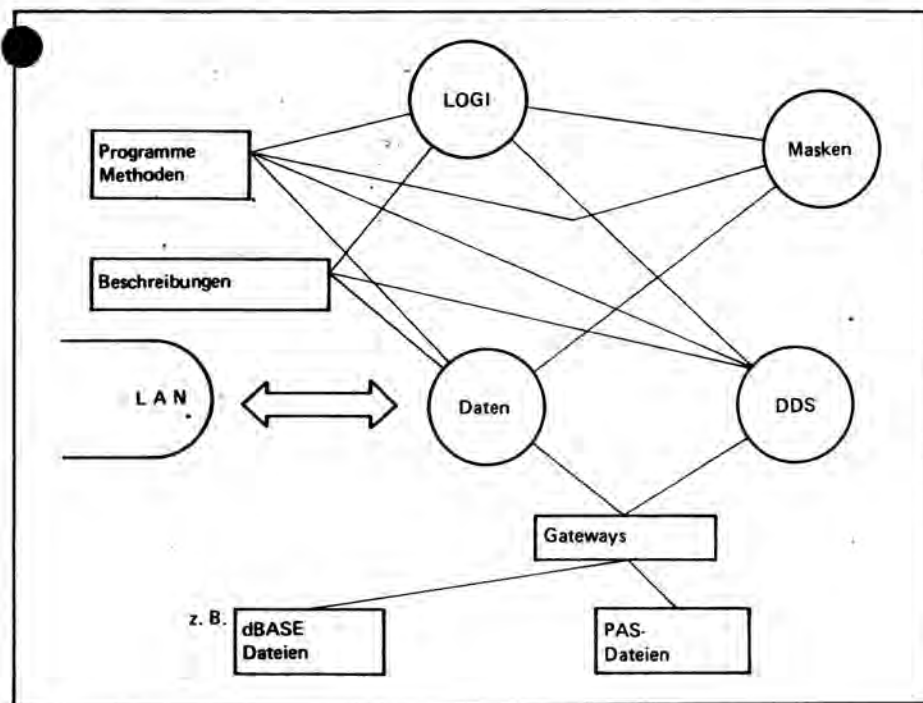


Abb. 1 Verbindung zwischen Programm (Methode), Methodenbeschreibung und dem DES über Masken

tung sind folgende Sprachelemente eines MBS definiert worden:

INSERT

Definition einer Methode;

UPDATE, DELETE

Manipulation;

LIST

Auskunft (Übersicht);

FIND

Auskunft (Führung eines Nutzers);

METHOD

Auskunft (über eine Methode);

IMPORT

Bereitstellung von Daten;

EXPORT

Ergebnisse der Methode.

Dateistruktur – MBASE

(ISW-Methodenbanksystem)

Zur definierten Beschreibung einer Methode wird eine feste Datenstruktur mit folgenden Komponenten benutzt, die im DES verwaltet wird:

1 – Name XXXXXXXX.YYY

2 – Langbezeichnung (60 Byte)

3 – CR-Classification Str(5)

4 – ISW-Classification Str(5)

5 – SL-Feld für Beschreibung und Dokumentation

6 – Stichworte zur Anwendung (VarStr)

7 – CLASS (Typ der Anwendung) (Char)

8 – Name der Eintrittsmaske bei Steuerung durch TINA

9 – INPUT (80 Byte)VarStr

10 – OUTPUT (80 Byte)VarStr

11 – IMPORT Name des Import-Programmes

12 – EXPORT Name des Export-Programmes

13 – TOOLS (80 Byte).

Der Verwendungsindex von Methoden (CLASS) ist eine semantische Klassifikation, die aus der allgemeinen semantischen Datenbeschreibung des DES übernommen wurde:

1 = Fakten – Recherche

2 = Soll-Ziel-Angaben

3 = Soll-Ist-Vergleich

4 = Prognose

5 = Logistic (Optimierung, Statistik u. ä.)

6 = Erklärungen (Dokumente u. ä.).

Methoden-Aktivierung

Der Ablauf einer Methoden-Aktivierung auf der Basis der Maskensteuerung ist wie folgt zu beschreiben:

1. Methoden-Auswahl

2. Datenauswahl bzw. -aufbereitung

3. Ausführung

4. Bereitstellen der Ergebnisse im Puffer

5. Auswahl der Darstellung (Liste, Graphik, Transfer)

6. Unterstützung bei der Ergebnisinterpretation.

Die Effektivität der MBS ist sehr stark an die Möglichkeiten der Datenbereitstellung gebunden. In der betrieblichen Realität sind alle den Reproduktionsprozeß widerspiegelnden Daten in der Regel vorhanden. Problematisch ist nur, daß sie auf der Basis heterogener Datenverwaltungssysteme in einer verteilten Umgebung existieren. Die Produktions-Planungs- und Optimierungsmethode nutzt z. B. als Eingangsdaten 21 verteilte Dateien, die in der Realität mit Sicherheit verteilt sind.

Deshalb sind die Arbeiten zu den verteilten Systemen für MBS von besonderer Bedeutung, denn ein MBS wird nur dann einer praktischen Nutzung zugänglich sein, wenn es möglich ist, die vorhandenen Daten dafür zu nutzen.

Z. Z. sind folgende Applikationen im Test bzw. in Vorbereitung:

- Statistik
- Produktionsplan-Optimierung
- Kostenvorausberechnung neuer Erzeugnisse
- Semantische Analyse in Texten
- Programm-System für den Angebotsingenieur eines Exportbetriebes
- Elektronische Ablage
- Versionsverwaltung in der Softwareproduktion
- Analysesystem ökonomischer Kennziffern.

Anschluß lokaler Datennetze an ein Wide Area Netz

(Prof. Dr. Martin Graef, Universität Tübingen)

So wie wir uns schon länger an den Warentransport und an den Personentransport gewöhnt haben und mit einer gewissen Selbstverständlichkeit die Sprachübertragung mittels Telefon benutzen, so beginnen wir uns allmählich auch an die Dienste der Datennetze zu gewöhnen. Im folgenden soll eine allgemeine Konzeption vorgestellt werden, wie sie in den letzten Jahren an der Universität Tübingen realisiert wurde.

Eine Besonderheit der Netzdienste ist, daß die Entwicklung unverhältnismäßig schneller als in den anderen Transportbereichen abläuft und die Benutzer voller Ungeduld auf die Realisierung warten. In gewissem Sinne ist aber die Situation auch schon überreif geworden, als die Schere zwischen Rechengeschwindigkeit auf der einen und der Datenübertragungsgeschwindigkeit auf der anderen Seite auch schon sehr weit auseinanderklaffte.

Als die Universität Tübingen sich im Herbst 1983 für eine Netzkonzeption entschied, war es noch sehr unklar, in welche Richtung die Entwicklung ihren Lauf nehmen würde. Entsprechend war die damalige Netzentscheidung eine echte Management-Entscheidung, d. h. eine Entscheidung aufgrund von Unkenntnis. Zurückblickend kann man sagen, daß wir nicht falsch gelegen haben – manches natürlich auch hätten besser machen können. Dafür waren allerdings auch die Voraussetzungen nicht immer gegeben. Die Netzdienste sind aber heute nicht mehr wegzudenken, und die gesammelten Erfahrungen tragen ihre Früchte.

Die Tübinger Netzkonzeption

Die Grundidee besteht in der Vorstellung, daß man

– von jedem Terminal aus zu jedem Rechner der Universität Tübingen grundsätzlich Zugriff haben sollte;

– anstelle eines Terminals auch einen PC einsetzen kann und mit diesem die

■ Die neueste Verbindung nach draußen ist eine 64 Kbd-ETHERNET-Brücke zum ETHERNET an der Universität Stuttgart. Auf dieser Grundlage werden alle Hochschulen des Landes Baden-Württemberg verbunden (BELWÜ). Der Tübinger Anschluß soll in den nächsten Jahren auf 10 MBd beschleunigt werden. Bei der Verbindung Stuttgart-Karlsruhe, die heute schon mit 10 MBd betrieben wird, denkt man an eine Erweiterung auf 140 MBd.

Ausblick

Die Aufgaben eines jeden Rechenzentrums – nicht nur der Hochschul-Rechenzentren – befinden sich seit einiger Zeit im Umbruch. Der reine Betrieb von Rechenanlagen tritt immer mehr in den Hintergrund. Die Hauptaufgabe in der Zukunft werden Ausbau und Betreuung von anspruchsvollen Datennetzen sein. Die hier anzubietenden Dienste beginnen, erste allmählich feste Formen anzunehmen. Hier können wir alle nur aus den Erfahrungen der anderen jeweils lernen. Aus dieser Sicht würde es mich sehr freuen, wenn auch die DDR-Hochschulen sich an EARN (European Academic Research Net) anschließen würden.

Datenaustauschkonzept für einen heterogenen Rechnerverbund

(Frank Bolz, Kombinat VEB Carl-Zeiss Jena)

■ Für den Datenaustausch im Kombinat VEB Carl Zeiss JENA wurde ein Datenpoolkonzept entwickelt, um unter Nutzung der DFÜ einen automatisierten Filetransfer für einen heterogenen Rechnerverbund zu realisieren.

■ Das Datenpoolkonzept ist dabei die einheitliche technologische und systemtechnische Schnittstelle zwischen den drei Ebenen der verteilten Datenverarbeitung: Kombinatsebene, Betriebsebene, Arbeitsplatzebene.

■ Der zentrale CZ-Datenpool wurde mit 300 MByte online-Datenraum eingerichtet und ist als „Durchgangsspeicher“ konzipiert. Für die Betriebsebene sind betriebliche Datenpools (BK-POOL) erforderlich, wobei der betriebliche POOL-Rechner gleichzeitig der DFÜ-Knoten zum zentralen Kombinaterechenzentrum ist.

Die drei Ebenen der verteilten Datenverarbeitung sind mittels Poolkonzept gekoppelt (Abb. 3).

■ Zum Datenpoolkonzept sind – POOL-Speicherung und -verwaltung und

– POOL-Datenaustausch entkoppelt. Systemtechnisch sind deshalb unter Beibehaltung der POOL-Organisation und ohne Änderung der DV-Anwendungssysteme folgende Möglichkeiten des Datenaustausches vorhanden:

- offline-DFÜ mit BC-Konverter (BC mit MB und DFÜ-Anschluß)
- online-DFÜ von ESER zu ESER, SKR/A6402 und PC/7150

– Netzkopplung von SKR/1840 zu SKR/1840, SKR/1420 und PC/1834

Für die online-DFÜ sind zwei Arbeitsweisen verfügbar:

– Abrufbetrieb: In diesem Modus ist auf einem Rechner eine Dialogbedienung aktiv, während der Partner im SERVER-Betrieb im Hintergrund läuft. In diesem Modus ist eine gezielte Dateiauswahl möglich.

– automatische Dateiversand: Sobald eine Datei in den POOL eingespeichert wird, erfolgt an Partner, die in diesem Modus arbeiten, eine sofortige Dateiübertragung. Diese Lösung arbeitet in beiden Richtungen.

■ Im Kombinaterechenzentrum wird über einen Auslieferungskatalog unabhängig vom DV-System für jeden Betrieb und jede Ausgabedatei festgelegt, welche Datenaustauschform benutzt wird (Prinzipdarstellung siehe Abb. 4).

■ Für den Datenaustausch von BK-POOL zur Arbeitsplatzebene sind folgende Kopplungen möglich

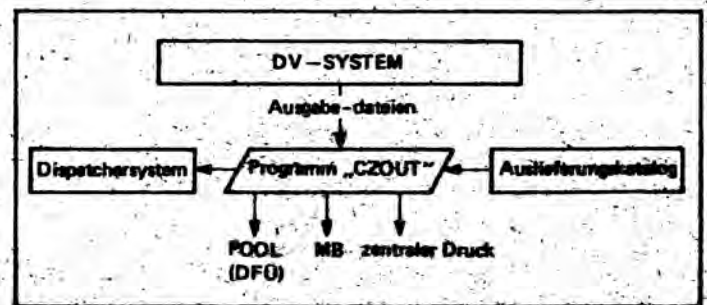
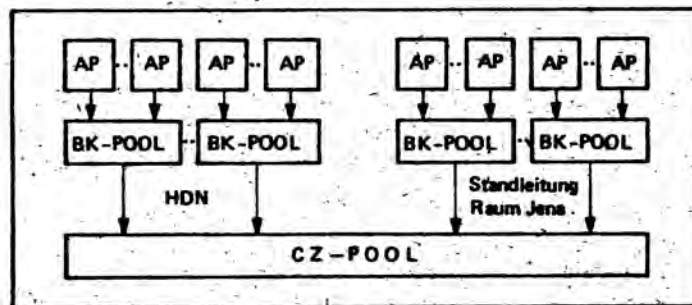
– offline-Datenträgeraustausch per Disketten

– Terminalanschluß für Terminals und PC's

– Netzkopplung für LAN oder Verbund. Für die Konvertierung und Rückkonvertierung von Disketten (SCP und DCP) wurden Softwarelösungen geschaffen, um eine effektive POOL-Ein- und -Ausgabe zu realisieren (z. B. zur direkten Ein- und Ausgabe von dBase-Dateien).

■ Mit diesem Poolkonzept wird für alle Aufgaben des Dateiaustausches eine einheitliche Lösung angeboten, unabhängig von der jeweils eingesetzten zentralen Hardware.

Abb. 3 Poolkonzept
Abb. 4 Prinzipdarstellung



Unser Programm zur Informationsverarbeitung 1989

Programmierung in Standard-BASIC

Von Ivo Machačka
Übersetzung aus dem Tschechischen
Etwa 192 Seiten, Broschur, etwa 12,00 M
Bestellangaben: ISBN 3-349-00300-1
676 140 3/Machačka, BASIC
Erscheint voraussichtlich im I. Quartal

Harter Kampf um „Weiche Ware“ Softwarestrategien der Monopole

Von Dr. sc. oec. Mathias Weber
und Dr. Heike Belitz
Etwa 176 Seiten, Broschur, etwa 10,80 M
Bestellangaben: ISBN 3-349-00451-2
676 271 3/Weber, Softwarestrat.
Erscheint voraussichtlich im II. Quartal

Das Betriebssystem DCP

Von Christian Mehnert
Herausgeber: Kombinat Robotron
Etwa 192 Seiten, Broschur, etwa 12,00 M
Bestellangaben: ISBN 3-349-00311-7
676 152 6/Betriebssystem DCP
Erscheint voraussichtlich im II. Quartal

Datenverwaltung mit REDABAS-3

Von Dipl.-Ing. Ursula Hempel
und Dipl.-Ing. Hans Loley
Etwa 192 Seiten, Broschur, etwa 12,00 M
Bestellangaben: ISBN 3-349-00439-3
676 259 6/Hempel, REDABAS-3
Erscheint voraussichtlich im II. Quartal

Der Personalcomputer EC 1834

Von einem Autorenkollektiv
Leitung Prof. Dr. Rolf Zeth
Herausgeber: Kombinat Robotron
Etwa 384 Seiten, Broschur, etwa 24,00 M
Bestellangaben: ISBN 3-349-00574-8
676 392 7/EC 1834
Erscheint voraussichtlich im III. Quartal

Wirtschaftsinformatik – Grundlagen

Von Prof. Dr. sc. oec. Gerd Goldammer,
Dr. sc. oec. Günter Huhn
und Prof. Dr. sc. Jochen Picht
Etwa 288 Seiten, Pappband, etwa 15,60 M
Bestellangaben: ISBN 3-349-00471-7
Erscheint voraussichtlich im IV. Quartal

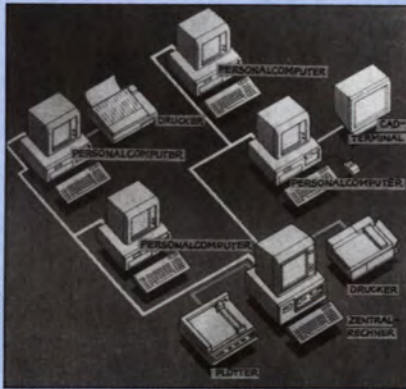
In BASIC effektiv programmieren Auch mit Kleinstrechnern

Von Prof. Dr. habil. Horst Völz
Etwa 192 Seiten, Broschur, etwa 12,00 M
Bestellangaben: ISBN 3-349-00570-5
676 377 5/Völz, BASIC-Programmierung
Erscheint voraussichtlich im IV. Quartal



Verlag Die Wirtschaft Berlin
Am Friedrichshain 22, Berlin, 1055

Management in lokalen Netzen



Lokale Rechnernetze (LAN) werden heute international vor allem auf den Gebieten Produktionsautomatisierung (Stichwort CIM) und Büroautomatisierung eingesetzt. Um einen sicheren Betrieb zu gewährleisten, sind neben den Funktionen, die den eigentlichen Zweck des Netzes bestimmen (z. B. File-Transfer, Mailboxdienst u. a.) auch solche Funktionen notwendig, die der Steuerung und Überwachung des Netzbetriebes dienen (Management). Derartige Funktionen können z. B. sein:

- Inbetriebnahme/Außerbetriebsetzung des Netzes,
- Registrieren und Überwachen des Status' wichtiger Netzparameter,
- Aufzeichnen und Verändern der Netzkonfiguration,
- Fehlerdiagnose/Wiederanlaufmechanismen.

Beim Entwurf von Konzepten für lokale Netze setzt sich die Tendenz fort, das OSI-Basis-Referenzmodell anzuwenden. Die Internationale Organisation für Standardisierung (ISO), die auch das OSI-Basis-Referenzmodell verabschiedet hat, beschäftigt sich seit einiger Zeit auch mit der Standardisierung von *Managementfunktionen*. So wurden z. B. folgende Managementfunktionen vorgeschlagen:

- Fehlermanagement (Fault Management),
- Abrechnungsmanagement (Accounting Management),
- Konfigurationsmanagement (Configuration Management),
- Leistungsmanagement (Performance Management),
- Sicherheitsmanagement (Security Management).

Im OSI-Basis-Referenzmodell werden drei Management-Kategorien definiert:

- Systemmanagement (systems management)
- Schichtenmanagement (layer management)
- Anwendungsmanagement (application management).

Schwerpunkt der Arbeiten bildet gegenwärtig das Systemmanagement. Es ist in der Anwendungsschicht (Schicht 7 des OSI-Basis-Referenzmodells) angeordnet und für die Verwaltung aller OSI-Ressourcen des Netzes verantwortlich. Die Kommunikation zwischen Managementeinrichtungen verschiedener Endsysteme (Rechner) erfolgt über Managementprotokolle, die die Vorschriften für den Ablauf der Kommunikation festlegen. Relativ weit fortgeschritten in der internationalen Standardisierung ist das CMIS-Projekt (Common Management Information Services).

Hartwig Kermas
 TU Dresden, Informatik-Zentrum