



---

# Handbuch der Informatik

herausgegeben von  
Prof. Dr. Albert Endres  
Prof. Dr. Hermann Krallmann  
Dr. Peter Schnupp

---

Band 7.1

---

# Standard- Architekturen für Rechner- kommunikation

---

von

Dr. Klaus-Jürgen Eckardt

Dr. Robert Nowak

Siemens AG, München

---

R. Oldenbourg Verlag München Wien 1988

---

**Dr. Klaus-Jürgen Eckardt**

Studium der Mathematik an der Universität München, 1973 Promotion zum Dr. rer. nat., Visiting Assistant Professor für Angewandte Mathematik an der University of Colorado. Seit September 1977 bei der Siemens AG in den Bereichen Softwareentwicklung für Datenfernverarbeitung und Kommunikation sowie Erarbeitung und Kommentierung internationaler Normen tätig.

**Dr. Robert Nowak**

Studium der Nachrichtentechnik an der Technischen Hochschule Stuttgart. Von 1963 – 1970 Student, Lecturer und Research Associate an der Stanford University, Radioscience Laboratory, 1967 Ph. D. Seit 1970 Hard- und Software-Entwickler bei der Siemens AG im Bereich Datentechnik.

Anschrift:

Siemens AG, Bereich Datentechnik  
Otto-Hahn-Ring 6, 8000 München 83

**CIP-Titelaufnahme der Deutschen Bibliothek**

**Handbuch der Informatik** : d. umfassende Darst. d. Informatik  
in Einzelbänden / hrsg. von Albert Endres ... – München ;  
Wien : Oldenbourg.

NE: Endres, Albert [Hrsg.]

Bd. 7.1. Eckardt, Klaus-Jürgen: Standard-Architekturen für  
Rechnerkommunikation. – 1988

**Eckardt, Klaus-Jürgen:**

Standard-Architekturen für Rechnerkommunikation / von  
Klaus-Jürgen Eckardt ; Robert Nowak. – München ; Wien :  
Oldenbourg, 1988

(Handbuch der Informatik ; Bd. 7.1)  
ISBN 3-486-20700-8

NE: Nowak, Robert:

© 1988 R. Oldenbourg Verlag GmbH, München

Das Werk einschließlich aller Abbildungen ist urheberrechtlich geschützt. Jede Verwertung außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Bearbeitung in elektronischen Systemen.

Lasersatz: Jens-Peter Oldenbourg, Grünwald

Gesamtherstellung: R. Oldenbourg Graphische Betriebe GmbH, München

ISBN 3-486-20700-8

---

# Inhalt

Vorwort der Herausgeber . . . . .	9
Vorwort der Autoren . . . . .	11

## A Einführung

---

1. Verteilte Datenverarbeitung . . . . .	15
1.1 Allgemeines . . . . .	15
1.2 Rechnernetze . . . . .	16
1.3 Aufgaben der verteilten Datenverarbeitung . . . . .	18
2. Architektur . . . . .	21
2.1 Strukturierung . . . . .	21
2.2 Verteilung und Zusammenarbeit . . . . .	23
3. Kommunikation . . . . .	31
3.1 Allgemeines . . . . .	31
3.2 Übertragungswege . . . . .	32
3.3 Netztopologien . . . . .	34
3.4 Aspekte der Datenübertragung . . . . .	37
3.5 Prozeduren . . . . .	40
3.6 Adressierung und Wegewahl . . . . .	46
3.7 Flußkontrolle . . . . .	49
3.8 Steuerung von Datenstationen . . . . .	53
3.9 Netzmanagement . . . . .	55
4. Normen und Standards . . . . .	59
4.1 Einführung . . . . .	59
4.2 Normierungsgremien . . . . .	61
4.3 Andere Organisationen . . . . .	62
4.4 ISO . . . . .	63
4.5 CCITT . . . . .	64

## B Standard-Architekturen

---

1. Einführung . . . . .	69
2. Das OSI-Referenzmodell . . . . .	71
2.1 Einführung . . . . .	71
2.2 Zweck des OSI-Referenzmodells . . . . .	72
2.3 Prinzipien und Begriffe . . . . .	74
2.4 Funktionsweise einer Schicht . . . . .	78

2.5	Die 7 Schichten . . . . .	88
2.5.1	Die Verarbeitungsschicht . . . . .	91
2.5.2	Die Darstellungsschicht . . . . .	93
2.5.3	Die Kommunikationssteuerungsschicht . . . . .	95
2.5.4	Die Transportschicht . . . . .	97
2.5.5	Die Vermittlungsschicht . . . . .	98
2.5.6	Die Sicherungsschicht . . . . .	102
2.5.7	Die Bitübertragungsschicht . . . . .	103
2.6	Die Struktur der Verarbeitungsschicht . . . . .	104
2.7	Die wichtigsten Netztypen und das OSI-Referenzmodell . . . . .	107
2.7.1	Das Telefonnetz . . . . .	108
2.7.2	Das paketvermittelnde Netz . . . . .	109
2.7.3	Das leitungsvermittelnde Netz . . . . .	110
2.7.4	Das ISDN-Netz . . . . .	111
2.7.5	Die lokalen Netze (LANs) . . . . .	112
2.8	Adressierung und OSI-Referenzmodell . . . . .	113
3	Funktionsnormen und Profile . . . . .	119
3.1	Einführung . . . . .	119
3.2	Transportfunktionen . . . . .	121
3.3	Relaisfunktionen . . . . .	123
3.4	Anwendungsfunktionen . . . . .	124
3.5	Anwendungserweiterungen . . . . .	125
3.6	Zeichensatzfestlegungen . . . . .	126
3.7	Nicht-OSI-Funktionen . . . . .	127
4.	OSI-Architekturmodell für das Systemmanagement . . . . .	129
4.1	Einführung . . . . .	129
4.2	Das Standard-Architekturmodell . . . . .	129
5.	Die Dienste des CCITT . . . . .	133
5.1	Einführung . . . . .	133
5.2	Relation der Dienste zum OSI-Referenzmodell . . . . .	134
5.3	Einzelne Dienste . . . . .	135
5.3.1	Teletex . . . . .	135
5.3.2	Telefax . . . . .	137
5.3.3	Mixed Mode . . . . .	138
5.3.4	Bildschirmtext . . . . .	138
5.3.5	Message Handling Systems . . . . .	139
6.	ARPANET und Herstellerarchitekturen . . . . .	145
6.1	Allgemeines . . . . .	145
6.2	Das ARPA-Netz . . . . .	148
6.2.1	Das Konzept . . . . .	148

6.2.2	Funktionen und Protokolle . . . . .	151
6.2.3	Vergleich mit dem OSI-Referenzmodell . . . . .	159
6.3	DNA . . . . .	161
6.3.1	Das Konzept . . . . .	161
6.3.2	Funktionen und Protokolle . . . . .	163
6.3.3	Vergleich mit dem OSI-Referenzmodell . . . . .	170
6.4	SNA . . . . .	171
6.4.1	Das Konzept . . . . .	171
6.4.2	Protokolle und Funktionen . . . . .	177
6.4.3	Vergleich mit dem OSI-Referenzmodell . . . . .	183
6.5	TRANSDATA . . . . .	184
6.5.1	Das Konzept . . . . .	184
6.5.2	Funktionen und Protokolle . . . . .	188
6.5.3	Vergleich mit dem OSI-Referenzmodell . . . . .	196
6.6	Büroarchitekturen . . . . .	197
6.6.1	Allgemeines . . . . .	197
6.6.2	SNA-Erweiterungen . . . . .	197
6.6.3	SBA . . . . .	199
7.	Kopplung und Migration . . . . .	201
7.1	Allgemeines . . . . .	201
7.2	Umwandlung von Protokollen . . . . .	204
7.3	Kopplung verbindungsorientierter und verbindungsloser Netze nach OSI . . . . .	204
7.4	Distributed End System Gateway . . . . .	206
7.5	Migration . . . . .	207
7.6	Eine Fallstudie: TRANSDATA . . . . .	208
7.6.1	NEATT und NEATE . . . . .	208
7.6.2	NEABX . . . . .	211

**C Anhang**

---

1.	Literaturempfehlungen . . . . .	217
2.	Literaturverzeichnis . . . . .	221
3.	Zitierte Normen . . . . .	226
4.	Abkürzungen . . . . .	232
5.	Deutsch-Englisches Glossar . . . . .	235
6.	Register . . . . .	239

---

---

## Vorwort der Herausgeber

Das *Handbuch der Informatik* versucht, das Gesamtgebiet der Informatik mit seinen Grundlagen, seinen Teilgebieten und seinen wichtigsten Anwendungen zusammenhängend darzustellen. Es informiert Lehrende und Lernende, DV-Praktiker und DV-Nutzer über Konzepte, Methoden und Techniken, deren grundlegende Bedeutung anerkannt ist, deren Nützlichkeit in der Praxis sich gezeigt hat; Grenzen und Entwicklungstendenzen eines Gebiets werden angesprochen.

Jeder Band des *Handbuch der Informatik* behandelt ein in sich abgeschlossenes Thema. Der Leser kann sich - auch unabhängig von den anderen Bänden des Handbuches - in das betreffende Gebiet neu einarbeiten, vorhandenes Wissen auffrischen oder im Sinne eines Nachschlagewerkes einzelne Themen aufspüren und vertiefen. Hierbei helfen Strukturierung und Typographie des Textes und die Hinweise auf weiterführende Literatur.

Es gibt nur wenige Gebiete der Datenverarbeitung, die gleichzeitig so komplex, schwierig *und* für die Praxis wichtig sind wie die Datenübertragung und ihre Standards.

Die *Komplexität* liegt an den harten Fakten (also der *Hardware*): während ein normaler Datenverarbeiter ausreichend damit beschäftigt ist, einen einzelnen Computer halbwegs zu verstehen und am Laufen zu halten, hat es der Netz-Implementierer und -Betreiber oft mit Hunderten zu tun - und das meist auch noch von verschiedenen Herstellern, in verschiedenen organisatorischen Umgebungen und für ganz unterschiedliche Anwendungen.

Die *Schwierigkeit* kommt vor allem daher, daß es für die Beherrschung dieser Komplexität zwar (notgedrungen) eine Menge Theorie gibt, aber diese Theorie nicht "elegant" ist und wahrscheinlich auch nie mehr werden kann. Das Gebiet der Programmiersprachen (zum Beispiel) hatte das Glück, daß ihm schon an seinen ersten Anfängen Wissenschaftler mit solidem mathematischen Können eine tragfähige Basis schufen, von der die Entwickler und Implementierer von Programmiersprachen noch heute zehren: Beschreibungsmittel für eine formale Syntax und (etwas später) für eine Formalisierung der Semantik.

Als die nächste Generation dieser Wissenschaftler sich dann der Rechnernetze annehmen wollte, fand sie leider schon so viele und so komplexe (s.o.) vor, daß sie nicht mehr viel an Eleganz, sondern nur noch an leidliche Ordnung des Chaos denken konnte. Wobei es beachtlich ist, was sie mit dem *ISO-Architekturmodell* trotz dieser Ausgangssituation noch leistete.

Über die *praktische Wichtigkeit* der Datenübertragung schließlich brauchen wir hier wohl keine weiteren Worte zu verlieren.

Über ein derartiges Gebiet einen Handbuchband zu schreiben, ist keine beneidenswerte Aufgabe. Sie kann nur von erfahrenen Praktikern mit solidem theoretischen Wissen gelöst werden, die in ihrer alltäglichen Arbeit genügend Komplexität erleben, daß sie vor ihr nicht mehr erschrecken. Sondern daß sie diese Komplexität uns anderen durchsichtig machen können.

Die Autoren dieses Bandes erfüllen diese Voraussetzungen. Für einen großen Hersteller setzen sie Theorie und Standards der Datenübertragung in praktische, weltweit installierte Lösungen um.

Ihre Erfahrung finden Sie in diesem Band.

A. Endres

H. Krallmann

P. Schnupp

---

## Vorwort der Autoren

Datenverarbeitung geschieht heute immer mehr "im Verbund", wobei Rechner unterschiedlichster Leistung, vom Arbeitsplatzrechner bis zum großen Zentralrechner, zusammenarbeiten. Oft sind diese Rechner und die daran angeschlossenen Datenstationen weit voneinander entfernt; die Kommunikation zwischen ihnen, d.h. der Austausch von Daten, ist ein wichtiger Aspekt der Zusammenarbeit.

Rechnernetze gibt es seit den siebziger Jahren; solange sie klein und auf eine Organisation beschränkt waren, konnten sie "aus einer Hand" erstellt werden: Ein Hersteller lieferte die Rechner und Kommunikationsgeräte und garantierte, daß seine Produkte "kompatibel" waren, d.h. zusammenarbeiten konnten.

Die Produkte verschiedener Hersteller konnten untereinander eine solche Kompatibilität nicht bieten; wegen der technischen Weiterentwicklung der Geräte mußten oft selbst beim gleichen Hersteller für die Zusammenarbeit aller Produkte besondere Aufwendungen erbracht werden. Mit dem Wachsen der Rechnernetze und vor allem dem aufkommenden Wunsch nach der Zusammenarbeit verschiedener Netze ist eine derartige Vielfalt nicht mehr tragbar; sie wäre vergleichbar mit lokalen Telefongesellschaften, bei denen man nicht über den Bereich des eigenen Amtes hinaus telefonieren könnte.

Die Zusammenarbeit von Systemen verschiedener Hersteller kann auf mehrere Arten ermöglicht werden:

- Zwischen den unterschiedlichen Systemen werden fallweise Anpassungen eingesetzt. Dies führt allerdings zu einer Vielzahl von Einzellösungen.
- Ein Hersteller versucht, seine "Spielregeln" der Kommunikation als Norm durchzusetzen; dies ist die Strategie, die der DV-Marktführer IBM mit seinem Regelwerk SNA (Systems Network Architecture) verfolgt.
- Die Fernmeldegesellschaften stellen neben den weltweit gültigen Normen für den Telefonverkehr auch Normen für die Datenkommunikation auf. Dies liegt nahe, da die physische Übertragung der Datensignale über den lokalen Bereich hinaus in den meisten Ländern nur bestimmten Telefongesellschaften, oft nur staatlichen Fernmeldebehörden gestattet ist. Ein Gremium für die Abstimmung der Telefongesellschaften untereinander existiert bereits seit langem: das Comité Consultatif International Télégraphique et Téléphonique (CCITT).
- Das internationale Normungsgremium ISO (International Organization for Standardization), dem auch das DIN angehört, legt wie in vielen industriellen Bereichen auch Normen für die Datenkommunikation fest. Diese Normung ist noch in vollem Gange; inzwischen haben sich die meisten DV-Hersteller verpflichtet, ihre Rechnerkommunikation nach den ISO-Normen auszurichten und einige bieten bereits Produkte nach diesen Normen an oder haben solche Produkte angekündigt. Da ISO und CCITT bei

der Normung der Datenkommunikation eng zusammenarbeiten, ist dieser Ansatz mit dem vorher genannten in Einklang.

Im folgenden werden zunächst die Probleme der Datenkommunikation allgemein geschildert und einige Begriffe und Prinzipien erläutert. Die weiteren Kapitel gehen auf die bei ISO festgelegten Normen für Rechnerkommunikation, relevante CCITT-Normen sowie einige herstellereigene Konzepte für verteilte DV ein.

Die technologischen Möglichkeiten der Datenverarbeitung und ganz besonders der verteilten DV wachsen heute explosionsartig. Fast täglich werden neue Anwendungen gefunden und Produkte veralten manchmal in Monaten. Wir haben uns bemüht, in dieser ständig wechselnden Szene Konzepte und grundlegende Ansätze herauszuarbeiten, die länger Bestand haben.

München, Oktober 1987

K.-J. Eckardt

R. Nowak

---

# A. Einführung

---

---

---

# 1. Verteilte Datenverarbeitung

## 1.1 Allgemeines

Zu Beginn war Datenverarbeitung im wesentlichen auf komplexe Berechnungen beschränkt, zunächst für militärische und wissenschaftliche Anwendungen, bald aber auch im kommerziellen Bereich, z.B. in der Versicherungsmathematik. Der Rechner war hier ein alleinstehendes "mathematisches Werkzeug". Heute sind dagegen im kommerziellen Bereich vor allem Anwendungen wichtig, bei denen nicht so sehr die Berechnung komplizierter Formeln, sondern Haltung und Manipulation großer Datenbestände im Vordergrund stehen. So können z.B. alle Lohndaten einer Firma, die Kundendaten einer Versicherung oder die Kontendaten einer Bank elektronisch gespeichert und bearbeitet werden. Solche Datenbestände müssen nun einer größeren Zahl von Sachbearbeitern zugänglich sein, die vorwiegend im "Dialog" arbeiten, d.h. Einzeldaten eingeben, abfragen oder verändern. Immer mehr handelt es sich um Daten, zu denen auch Mitarbeiter im Außendienst und am Kundenschalter zugreifen müssen.

Zugriff zu Daten

Schon früh wurden für solche Zugriffe entfernte Datenstationen an die Verarbeitungsrechner angeschlossen. Solche Datenstationen werden auch heute noch in großer Zahl eingesetzt; sie bestehen normalerweise aus Bildschirm und Tastatur und sind reine Ein/Ausgabegeräte ohne eigene Verarbeitungsleistung und Speicherung.

Datenstationen

Mit dem Vordringen der Datenverarbeitung in immer neue Anwendungsbereiche brauchen Sachbearbeiter für komplexe Aufgaben mit mehreren Bearbeitungsschritten eine "Bedienerführung", z.B. durch Vorgabe von Formularen oder durch gezielte Anforderung von Eingaben. Eine solche Bedienerführung ist "aus der Ferne" nicht vernünftig möglich, da die Kosten für die Übertragung der Dialogdaten zu hoch und die Antwortzeiten wegen der begrenzten Übertragungsgeschwindigkeit zu lang sind. Dazu kommt, daß in einem größeren Unternehmen viele Daten gar nicht überall benötigt werden, sondern nur beispielsweise in einer Abteilung von Interesse sind; eine Hin- und Herübertragung dieser Daten zu einer Zentrale wäre für die eigentliche Aufgabe nutzlos.

Bedienerführung

Beide Aspekte begünstigen eine Verteilung von Rechenleistung und Datenhaltung auf "Abteilungsrechner" oder sogar die Arbeitsplätze der Sachbearbeiter, was heute kostengünstig möglich ist. Für die Zusammenarbeit innerhalb der Gesamtorganisation muß jedoch die Konsistenz der Daten gewährleistet sein und der Bezug auf gemeinsame Datenbestände bleibt bestehen. Damit ist verteilte Datenverarbeitung gegeben, bei der mehrere Rechner in einem Netz zusammenarbeiten müssen.

Verteilung/Zusammenarbeit

Während sich noch vor kurzem Datenkommunikation fast nur zwischen einem Menschen an einer Datenstation und einem Programm in einem Rechner abspielte, steht bei verteilter Datenverarbeitung immer mehr die Kommunikation zwischen Programmen im Vordergrund. Dabei bleibt es nicht bei der Vernetzung der Rechner einer Organisation: Weil immer mehr Außenbeziehungen (Bestellungen, Rechnungsstellungen, Zahlungen) mit Hilfe der Datenverarbeitung abgewickelt werden, wird auch der direkte Datenaustausch zwischen verschiedenen Organisationen, also die Kopplung unterschiedlicher Rechnernetze, immer wichtiger.

Integration

Durch billigere Rechenleistung und neue Datennetze bietet die verteilte DV immer neue Möglichkeiten und findet Anwendung in allen Bereichen, in denen Informationen gehalten oder verarbeitet werden; vor allem im Büro können immer mehr Aufgaben durch Rechner unterstützt werden, und auch bei Konstruktion und Fertigung setzt sich die Datenverarbeitung durch. Das Stichwort heißt "Integration": Alle Aktivitäten eines Unternehmens - Planung, Verwaltung, Vertrieb, Kundenbetreuung, Lagerhaltung, Konstruktion und Fertigungssteuerung - sollen koordiniert werden und sich auf gemeinsame, konsistente Daten stützen.

Voraussetzung für die Ausschöpfung aller Möglichkeiten der Datenverarbeitung in einem solchen integrierten Umfeld ist eine leistungsfähige Kommunikation. Als Vorbild dafür kann das weltweite Telefonnetz genommen werden, in dem jeder Teilnehmer jeden anderen erreichen kann und lokale Kommunikation in Nebenstellenanlagen (also kleinen privaten Netzen) möglich ist.

Leistungsfähiges  
Kommunikations-  
system

Das Telefonnetz überträgt nur Schallwellen: Die eigentliche Kommunikation, also die Übermittlung von Sinn-Inhalten, müssen die Teilnehmer selbst leisten und sie müssen sich dazu abstimmen - schon alleine dadurch, daß sie die gleiche Sprache sprechen. Bei der Koordinierung von DV-Aktivitäten muß das Kommunikationssystem diese Abstimmung unterstützen und dazu erheblich mehr Leistungen erbringen als nur die reine Übertragung von Bitströmen. Diese Leistungen und die Organisation ihres Zusammenspiels - also die Architektur eines Kommunikationssystems - sind das Thema dieses Buches.

### 1.2 Rechnernetze

In einem System der verteilten Datenverarbeitung arbeiten mehrere Rechner - oft sehr unterschiedlicher Leistung - zusammen. Hierbei werden Rechenleistung und Datenhaltung nach mehreren Gesichtspunkten verteilt:

Leistungssteigerung

- Die Aufteilung der Aufgaben eines Unternehmens auf mehrere Rechner wird oft dadurch erzwungen, daß die Summe der erforderlichen Daten und Rechenleistungen von einem Rechner nicht mehr bewältigt werden kann.

Dabei genügt eine bloße Vervielfachung nicht: da viele Aufgaben in einem Unternehmen koordiniert abgewickelt werden müssen, müssen die Rechner vernetzt werden.

- Ordnet man Rechner den Organisationseinheiten des Unternehmens zu, so ergibt sich eine klare Abgrenzung der Verantwortlichkeiten; die unwirtschaftliche Duplizierung von Daten und Überschneidung von Aktivitäten, mit unvermeidbaren Inkonsistenzen, unterbleibt. Abbildung der Organisation

- Die Anforderungen an das DV-System einer Organisation ändern sich ständig. Der Arbeitsumfang steigt, es sind neue Aufgaben zu lösen, Sachbearbeiter kommen hinzu oder verändern ihren Arbeitsplatz. Diese Änderungen werden am einfachsten und kostengünstigsten in einem Netz beherrscht, in dem kleine, modulare Einheiten (Rechner, Geräte) nach Bedarf hinzugefügt oder verschoben ("umkonfiguriert") werden können. Flexibilität

- Speicherung und Verarbeitung von Daten sind heute billiger als deren Übertragung über größere Entfernungen; dieser Kostenvorteil wird sich in Zukunft noch vergrößern. Es ist also wirtschaftlich, Rechenleistung und Datenhaltung jeweils "vor Ort" dort zu erbringen, wo sie unmittelbar benötigt werden, und den Datentransfer auf die notwendigsten Informationen - wo möglich, in zusammengefaßter Form - zu beschränken. Andererseits können oft Rechenwerke, Plattenspeicher oder teure Ein/Ausgabe-Geräte wie z.B. Laserdrucker im lokalen Bereich, wo die Kommunikationskosten niedrig sind, von mehreren Anwendern genutzt werden. Kosten

Ein Rechnerverbund kann nun unter mehreren Aspekten gesehen werden:

- Beim *Lastverbund* kommt es vor allem auf die Erhöhung der Gesamtleistung des Systems an, die durch Abwicklung gleichartiger Aufgaben an mehreren Stellen gleichzeitig erreicht wird. Im einfachsten Fall ist die Aufteilung der Last auf die einzelnen Rechner statisch und fest vorherbestimmt, indem z.B. bestimmte Benutzer immer bestimmten Rechnern zugeordnet werden. Ein Lastverbund ist aber dann am wirkungsvollsten, wenn die Last während des Betriebes automatisch immer so auf die Rechner verteilt wird, daß alle gleichmäßig ausgelastet sind. Lastverbund

- Durch Vervielfachung wird gleichzeitig ein *Sicherheitsverbund* erreicht, bei dem bei Ausfall eines Rechners dessen Aufgaben von anderen Rechnern übernommen werden können. Beim statischen Lastverbund muß dazu ein Operateur eingreifen, beim dynamischen Lastverbund rekonfiguriert sich das System selbst. Sicherheitsverbund

- Beim *Funktionsverbund* werden unterschiedliche Aufgaben auf spezialisierte Rechner oder gemeinsam genutzte Geräte verteilt. Eine besondere Ausprägung ist der *Programmverbund*, bei dem bestimmte, aufwendige Programme nur in einem Rechner ablaufen, aber auch von anderen Rechnern aus angestoßen werden können. Funktionsverbund  
Programmverbund

Datenverbund

• Beim *Datenverbund* werden Datenmengen, die vorwiegend an einem Rechner gebraucht werden, zu denen aber gelegentlich auch von anderen Rechnern aus zugegriffen werden soll, jeweils dort gehalten, wo sie hauptsächlich verwendet werden, um Übertragungskosten zu sparen. Zusätzlich kann, durch Speicherung der gleichen Daten an mehreren Orten, ein Sicherheitsverbund erreicht werden, der gegen Datenverlust bei Ausfall eines Speichers schützt.

Leistungssteigerung  
Kostenminderung  
Ausfallsicherheit

In der Praxis haben diese verschiedenen Verbundaspekte, die Leistungssteigerung, Kostenminderung und Ausfallsicherheit betreffen, oft vergleichbares Gewicht. Der ideale Verbund ist dann erreicht, wenn sich den Partnern einer Kommunikation - einem Benutzer an einer Datenstation oder einem Programm - das gesamte Netz wie ein einziges System darstellt, in dem der Ort des anderen Partners unsichtbar und unwichtig ist, und in dem alle Betriebsmittel global vom System selbst verwaltet werden.

### 1.3 Aufgaben der verteilten Datenverarbeitung

In der verteilten Datenverarbeitung gibt es wie bei einzelnen Rechnern prinzipiell zwei Arbeitsweisen:

Stapelbetrieb  
(Batch)

• Im Stapelbetrieb (Batch Processing) wird ein Programm mit Eingabe von Daten gestartet und es läuft dann als "Auftrag" oder "Job", bis es die Ergebnisdaten zurückgeben kann. Eine solche Arbeitsweise, früher die einzig mögliche, ist auch heute noch für Anwendungen üblich, bei denen vorgegebene Daten nach festen Regeln bearbeitet werden - z.B. komplexe wissenschaftliche Berechnungen oder Buchhaltungsabschlüsse. Während hier früher die Daten lokal beim Rechner ein- und ausgegeben wurden, können heute im Fernstapelbetrieb (Remote Batch) Ein- und Ausgabedaten über Leitungen transportiert werden.

Fernstapelbetrieb  
(Remote Batch)

Dialogbetrieb

• Im Dialogbetrieb (Interactive Processing) wird eine Aufgabe schrittweise erledigt, wobei im allgemeinen ein Mensch an einem Terminal jeden Schritt anstößt und auf dessen Ergebnis wartet, um daraus den nächsten Schritt abzuleiten. Im einfachsten Fall arbeitet der Benutzer dabei mit einem eigenen Rechner (Personal Computer, PC). In den 70er Jahren, als ausreichende Rechenleistung und Speicherkapazität noch nicht kostengünstig am Arbeitsplatz verfügbar war, bot der Teilnehmerbetrieb (Time Sharing) die Möglichkeit der gemeinsamen Nutzung eines Rechners durch mehrere Benutzer. Dabei sorgt eine zyklische Zuteilung fester Zeitschnitte an die verschiedenen Benutzer dafür, daß auch lang laufende Programme den Rechner nicht monopolisieren. Obwohl beim starken Preisverfall und Leistungsanstieg der PCs die Bedeutung großer Timesharing-Rechenzentren gesunken ist, kann die Mehrfachnutzung von Rechnern im

Teilnehmerbetrieb

Abteilungsbereich durch Teilnehmerbetrieb immer noch merkliche Kostenvorteile bringen.

Eine sehr wichtige Form des Dialogbetriebes ist der Teilhaberbetrieb, bei dem mehrere Benutzer mit dem gleichen Programm – z.B. einem Platzbuchungssystem – arbeiten. Während der Teilnehmerbetrieb die Mehrfachnutzung eines Rechners zum Ziel hat und die einzelnen Teilnehmer meist unabhängig voneinander arbeiten, haben im Teilhaberbetrieb die Benutzer die gleiche Aufgabe und beziehen sich dabei auf einen gemeinsamen Datenbestand, der in einer Datenbank gehalten wird. Meist werden in dieser Betriebsart Transaktionen abgewickelt; sie wird deshalb im Englischen als "Online Transaction Processing" (OLTP) bezeichnet. Eine Transaktion ist ein Vorgang (z.B. ein Transfer zwischen zwei Bankkonten), der als Ganzes abgewickelt werden muß, damit alle betroffenen Daten konsistent bleiben (HÄU79). Je nach Zahl der Benutzereingaben für einen Vorgang unterscheidet man zwischen Einschritt- und Mehrschritt-Transaktionen.

Teilhaberbetrieb

Transaktionen

Um die vielfältigen Aufgaben in einem Rechnernetz zu bewältigen, müssen nun verschiedene Funktionen erbracht werden:

- Die Koordinierung von entfernt voneinander ablaufenden Aktivitäten ist nur durch Nachrichten möglich. Dies können Ein- oder Ausgabedaten im Dialog- oder Fernstapelbetrieb, Benutzerkommandos und Systemmeldungen oder Nachrichten zur Systemverwaltung sein. Sie werden zwischen einem Menschen und einem Programm, zwischen zwei Programmen oder zwischen zwei Menschen ausgetauscht. Die sichere Übertragung dieser Nachrichten muß gewährleistet sein. Bei gesprochenen Nachrichten am Telefon haben gelegentliche Übertragungsstörungen keine große Bedeutung für die Verständigung der Partner; bei der Datenübertragung hingegen darf kein einziges Bit verfälscht sein, vom Verlust ganzer Nachrichten oder Zustellung in falscher Reihenfolge ganz zu schweigen.

Sichere Übertragung der Daten

- Da die Partnerbeziehungen in einem Netz im allgemeinen oft wechseln, müssen Vermittlungsaufgaben mit (netzweiter) Adressierung der Partner und Wegfindung für die zu übertragenden Daten gelöst werden.

Vermittlung

- Oft sind die Nachrichten Dateien, größere, strukturierte Datenmengen, deren Transfer (File Transfer) besondere Koordinierungsleistungen erfordert.

Dateitransfer

- Beim Transaktionsbetrieb ist ein leistungsfähiges Systemprogramm, ein sogenannter Transaktionsmonitor, notwendig, um den gemeinsamen Datenbestand bei konkurrierenden Änderungen auch über Systemabstürze hinweg immer konsistent zu halten.

Transaktionsmonitor

- Im Verbundbetrieb wird eine Aufgabe oft nicht in dem Rechner ausgeführt, in dem sie anfällt. Dann ist ein Job-Transfer notwendig. Hier sind oft verwickelte Auftragsbeziehungen zu beherrschen, bei denen die Aufgabe von einem Rechner aus angestoßen und in einem zweiten ausgeführt

Job-Transfer

wird, wobei schließlich die Ergebnisse an einem dritten Rechner ausgegeben werden.

- Fern-Prozeduraufruf • Besonders im lokalen Bereich kann es vorteilhaft sein, Teil- oder Unteraufträge an andere Rechner weiterzugeben. Dazu ist die Funktion des Fern-Prozeduraufrufes (Remote Procedure Call) zu erbringen.
- Elektronische Post • Bei der Büro-Automatisierung ist der Austausch von Briefen wichtig. Diese sind durch Electronic Mail-Systeme zu verteilen und auf unterschiedlichen Ausgabegeräten einheitlich darzustellen.
- Archivierung • Da immer mehr Daten elektronisch gespeichert werden, gewinnt eine effiziente Archivierung großer Datenmengen an Bedeutung.
- Netzmanagement • Es sind umfangreiche Administrationsdienste notwendig, um das Funktionieren des Gesamtsystems zu gewährleisten und Umkonfigurationen leicht zu ermöglichen.

---

## 2. Architektur

### 2.1 Strukturierung

Bereits die skizzenhafte Liste von Funktionen in Kapitel 1.3 läßt ahnen, daß für die Datenkommunikation umfangreiche und komplexe Hard- und Software notwendig sind. Eine solche Komplexität läßt sich nur dadurch beherrschen, daß das Gesamtsystem in Teile zerlegt wird, die nach einem übergeordneten Konzept zusammenarbeiten, dabei aber so klar voneinander getrennt sind, daß einzelne Teile geändert werden können, ohne daß dies Änderungen in anderen Teilen notwendig macht.

Modularisierung

Nun bilden alle diese Teile oder Moduln ein Gesamtsystem und sollen für die Ziele dieses Systems zusammenarbeiten. Die Regeln der Zusammenarbeit eines Moduls mit einem anderen - also z.B. Art und Abfolge von ausgetauschten Signalen oder Struktur und Bedeutung gemeinsam genutzter Datenbereiche - bezeichnet man in ihrer Summe als Schnittstelle zwischen den beiden Moduln. Ein Modul ist durch seine Funktionen und seine Schnittstellen zu den anderen Moduln des Systems bzw. zur "Außenwelt" vollständig beschrieben.

Schnittstelle

Es liegt nahe, ein System nach Funktionen oder Funktionsgruppen aufzuteilen. Die einzelnen Moduln sollten dabei so gewählt werden, daß ihr Zusammenwirken, also die Gesamtheit der Schnittstellen zwischen ihnen, möglichst einfach und übersichtlich wird. Ein besonders klares Ordnungsprinzip, das sich auch in modernen Betriebssystemen bewährt hat, ist die Schichtung. Hier werden die Funktionen so aufeinander aufgebaut, daß in jeder Schicht die Funktionen der darunterliegenden Schicht als Dienste in Anspruch genommen werden und für die darüberliegende Schicht Dienstleistungen erbracht werden. In diesem Konzept gibt es Schnittstellen nur zwischen benachbarten Schichten.

Funktionsschichtung = Hierarchie von Diensten

Eine solche Schichtung ist für Kommunikationsfunktionen von ISO als Norm festgelegt worden. Im OSI-Referenzmodell für offene Kommunikation (Open Systems Interconnection) wurden die Kommunikationsfunktionen 7 Schichten zugeordnet. Abb. 2.1-1 skizziert, welche Dienste jede Schicht erbringt; dieses Modell wird in Abschnitt B ausführlich dargestellt. Bereits in der kurzen Übersicht wird deutlich, wie die Funktionen aufeinander aufbauen: Der Schicht 1 werden all die Funktionen zugeordnet, die für die physische Übertragung der Daten auf einer Leitung zwischen zwei benachbarten Systemen notwendig sind. Die Instanzen der Schicht 2, die für die Fehlerfreiheit dieser Daten zu sorgen haben, können also davon ausgehen, daß die Daten physisch übertragen worden sind. Die Instanzen der Schicht 3 wiederum, die die Daten im Netz "weiterreichen", können sich darauf verlassen, daß die Daten frei von Übertragungsfehlern

OSI-Referenzmodell für offene Kommunikation

## A 2. Architektur

---

7	Verarbeitungsschicht (Application Layer)	Dienste für den Anwender, z.B. - File Transfer - Job Transfer - Elektronische Post
6	Darstellungsschicht (Presentation Layer)	Darstellung der Daten nach festgelegten Syntaxen sowohl für die Anwendung als auch für den Transfer zum Partner
5	Kommunikationssteuerungsschicht (Session Layer)	Organisation und Synchronisation des Dialoges zwischen den Teilnehmern auf der Basis einer "Session" (Sitzung)
4	Transportschicht (Transport Layer)	Zuverlässiger Datentransfer zwischen den Teilnehmern ("Ende-zu-Ende")
3	Vermittlungsschicht (Network Layer)	Datentransfer zwischen den Systemen, in denen die Teilnehmer residieren
2	Sicherungsschicht (Data Link Layer)	Zuverlässige Datenübertragung auf den Leitungen im Netz
1	Bitübertragungsschicht (Physical Layer)	Physische Übertragung der Daten auf den Leitungen im Netz

Abb. 2.1-1: OSI-Referenzmodell für offene Kommunikation

---

sind, weil die Instanzen der Schicht 2 bereits dafür gesorgt haben. So kann jede weitere Schicht sich auf immer umfangreichere Dienste der darunterliegenden Schichten abstützen.

In einem komplexen System führt allerdings meist die "vertikale" Aufteilung der Funktionen in Schichten noch nicht zu genügend kleinen und übersichtlichen Modulen; die Funktionen der einzelnen Schichten sind so vielfältig, daß eine zusätzliche "horizontale" Strukturierung notwendig ist. Oft ist es nun möglich, innerhalb einer Schicht Einzeldienste zu identifizieren, die unabhängig voneinander sind und daher untereinander keine Schnittstellenbeziehungen haben. Ein sauber strukturiertes System erlaubt

"Horizontale"  
Strukturierung

die Zusammenarbeit jedes dieser Einzeldienste mit denen der Nachbarschichten; da aber die freie Zusammenstellung von Dienstvarianten zu sehr vielen verschiedenen Kombinationen führt, werden in der Praxis meist nur einige aus bestimmten Einzeldiensten zusammengestellte Dienstsäulen unterstützt.

Dienstsäulen

Wenn die internen Strukturen der einzelnen Moduln nach außen nicht sichtbar sind, so können sie - solange die Schnittstellen zu den anderen Moduln unverändert bleiben - geändert werden, ohne andere Systemteile zu beeinflussen. Dadurch ist es möglich, in Teilen des Systems mit minimalem Aufwand neue Technologien einzuführen. Insbesondere kann frei gewählt (und verändert) werden, ob die Funktionen eines Teilsystems in Hard- oder Software erbracht werden sollen - ein wichtiger Gesichtspunkt bei der heutigen Tendenz, immer mehr Funktionen in hochintegrierten Schaltkreisen zu realisieren. Ein weiterer Vorteil eines modularisierten Systems ist, daß die einzelnen Moduln unabhängig voneinander entwickelt und (mit entsprechenden Schnittstellensimulatoren) getestet werden können.

Vorteile der Modularisierung

Die Modularisierung und besonders die Schichtung hat allerdings auch ihren Preis: Da das geschichtete System seine Aufgaben für den allgemeinen Fall erbringen soll, beinhaltet der einheitliche Ablauf oft Aufwand, der für Einzelaufgaben unnötig ist; er ist damit nicht so effizient, wie es eine "handgestrickte", auf den jeweiligen Einsatzfall optimierte Lösung wäre. Dieser Nachteil wird - in einer Zeit rapide wachsender Hardwareleistung bei dafür sinkenden Kosten - zumindest bei größeren Systemen in Kauf genommen: Nur ein strukturierter Ansatz, der ein "Baukastenprinzip" beim Systemaufbau erlaubt, ist für die immer zu erwartenden Änderungen und Erweiterungen des Systems genügend flexibel und hält den Aufwand dafür überschaubar.

Preis der Modularisierung

## 2.2 Verteilung und Zusammenarbeit

In Verarbeitungsrechnern höherer Leistung werden Aufgaben oft, zur Steigerung von Leistung und/oder Verfügbarkeit des Gesamtsystems, auf mehrere Prozessoren verteilt. In solchen Multiprozessor-Rechnern haben alle Prozessoren direkten Zugang zum gemeinsamen Hauptspeicher, sie sind fest gekoppelt. Die Prozessoren koordinieren hier ihre Abläufe über gemeinsame ("globale") Variable oder Tabellen, ein gemeinsames Betriebssystem verwaltet alle Betriebsmittel.

Fest gekoppelte Prozessoren

Komplexer sieht die Sache bei unabhängigen Prozessoren mit individuellen Betriebssystemen aus, die lose gekoppelt sind, d.h. keinen gemeinsamen Arbeitsspeicher haben, sondern ihre Aufgaben nur über Nachrichten koordinieren können. Hier sind besondere Absprachen bezüglich der Übermittlung dieser Nachrichten notwendig.

Lose gekoppelte Prozessoren

Beispiele sind:

- Übereinstimmung von Spannungspegel und Übertragungsgeschwindigkeit bei Sender und Empfänger;
- Absprachen über die Sicherungsinformation, die den Nachrichten zuge-  
setzt wird und über die Art, wie Übertragungsfehler gemeldet und korri-  
giert werden;
- Vereinbarungen über die Struktur von Adressen;
- Absprachen über die Art, wie die Darstellung von Text, also z.B. Zeilen-  
umbruch oder Hervorhebungen, gesteuert wird.

Protokoll

Die Summe der Absprachen für die koordinierte Erbringung eines bestimmten Dienstes über mehrere Rechner hinweg, wird als *Protokoll* für diesen Dienst bezeichnet. Ein solches Protokoll regelt den Austausch von Informationen zwischen den an diesem Dienst beteiligten Instanzen. Diese Informationen werden - mit Hilfe der Dienste der darunterliegenden Schichten - manchmal in eigenen Steuernachrichten übermittelt, meist aber den Nachrichten höherer Schichten als Nachrichtenkopf oder Header vorangesetzt. Das Protokoll regelt Darstellung (Syntax) und Bedeutung (Semantik) der ausgetauschten Informationen und deren zeitliche Abfolge. Dieses Prinzip ist in Teil B, Kapitel 2.3 ausführlicher erläutert.

Header

Deadlock/Livelock

Bei der Definition eines Protokolls muß darauf geachtet werden, daß es nicht zu Blockierungszuständen kommen kann; ein solcher Zustand kann zwei Formen annehmen: Beim Deadlock kann keiner der beiden Partner mehr weiterarbeiten, weil jeder auf eine Aktion des anderen wartet und das Protokoll keine Sprachmittel hat, um diesen Wartezustand zu beenden. Beim Livelock sind beide Partner so miteinander verklemt, daß sie zwar weiterhin Signale austauschen, aber trotzdem keinen realen Arbeitsfortschritt erzielen.

In einem geschichteten System, in dem Dienste auf jeder Schicht erbracht werden, gibt es für jede Schicht (mindestens) ein Protokoll. Ein solches Protokoll muß es auch zwischen den Endbenutzern des Kommunikationssystems geben, die sich letztendlich über die Bedeutung der ausgetauschten Daten einig sein müssen. Solche Endbenutzer sind Menschen oder Rechnerprogramme.

Datenstationen

Ein Mensch braucht zur Kommunikation mit einem Rechner eine Datenstation. Eine solche Station kann direkt am Rechner angeschlossen sein oder sogar ein integraler Bestandteil dieses Rechners sein, wie dies bei Personal Computer (PC) oder Arbeitsplatzrechner (Work Station) der Fall ist. Hier wird jedes eingegebene Zeichen sofort zum Rechner übertragen, dort an das empfangende Programm weitergegeben und gleichzeitig auf dem Bildschirm als "Echo" angezeigt. Die Übertragung wird als fehlerfrei angenommen, so daß lediglich Absprachen über den elektrischen Anschluß des Gerätes sowie die Bildschirm-Darstellung und Tastaturbelegung notwendig sind.

Bei Rechnern, die mehrere Datenstationen unterstützen können, werden diese Stationen oft in einiger (manchmal sogar großer) Entfernung über eine Leitung angeschlossen. Hier kann zum einen eine fehlerfreie Übertragung nicht mehr garantiert werden; es ist also ein Sicherungs-Protokoll (früher als Prozedur bezeichnet) notwendig. Zum anderen ist wegen der sehr viel geringeren Übertragungsgeschwindigkeit als bei Direktanschluß eine zeichenweise Übertragung mit Echo nicht mehr akzeptabel: Die Daten werden also nachrichten- (oder "block"-)weise übertragen. Ebenso wird auch bei der Kommunikation zwischen zwei Programmen verfahren; hier müssen noch zusätzlich Dateitransfer-Protokolle abgesprochen sein.

Datenübertragungs-  
Prozedur

Die Art der Zusammenarbeit zweier Instanzen kann man nun nach zwei verschiedenen Strategien organisieren:

- Beim hierarchischen (Master/Slave-)Prinzip wird der Gesamtablauf von einer der beiden Instanzen gesteuert, die als Leitinstanz (Master) der anderen Befehle gibt; die Folgeinstanz (Slave) kann dabei nicht von sich aus aktiv werden, auch wenn sie eine Information abzugeben hätte, sondern sie muß die Aufforderung des Masters abwarten.

Master/Slave

Bei hierarchischer Organisation sind die Abläufe meist übersichtlich und gut beherrschbar. Oft ist das Master/Slave-Prinzip durch die Anwendung vorgegeben, besonders wenn ein Master mehrere Slaves steuern soll, wie es z.B. bei der Kopplung mehrerer Datenstationen an einen Rechner der Fall ist. Hier wirkt sich auch noch kostengünstig aus, daß die mehrfach vorhandenen Slaves einfacher und billiger sein können als der Master.

Nachteilig ist, daß die Slaves untereinander nur mühsam und aufwendig, durch Vermittlung eines oder mehrerer Master, zusammenarbeiten können. Für die Kopplung von Rechnern, bei denen im allgemeinen nicht ohne weiteres ersichtlich ist, wer die bevorrechtigte Leitinstanz sein soll, erscheint eine Master/Slave-Beziehung unnatürlich; sie wird hier daher normalerweise nur eingesetzt, wenn einzelne Rechner in ein bestehendes hierarchisches Netz eingebracht werden sollen.

- Bei gleichberechtigter (Peer-to-Peer-)Kommunikation kann jede der beiden Partnerinstanzen von sich aus aktiv werden und Nachrichten an die andere schicken. Wegen der dabei möglichen Überschneidungen ist diese Art der Zusammenarbeit schwieriger zu koordinieren als die hierarchische. Sie ist jedoch flexibler, da sie bei beliebigen Netzstrukturen die Zusammenarbeit "jedes mit jedem" erleichtert.

Peer-to-Peer

Für größere Unternehmen ist oft eine dreistufige Struktur des Rechnernetzes vorteilhaft; in solchen Netzen gibt es

Netzstruktur

- "Zentrale" Rechner, in denen Daten gehalten und manipuliert werden, die das gesamte Unternehmen oder einen seiner Standorte betreffen;
- "Abteilungsrechner" für die Speicherung und Verarbeitung der in einer Abteilung gemeinsam genutzten Daten;

– "Arbeitsplatzrechner" für die Aufgaben einzelner Mitarbeiter oder Projektgruppen.

Dabei sind im allgemeinen mehrere Arbeitsplatzrechner mit einem Abteilungsrechner und wiederum mehrere Abteilungsrechner mit einem zentralen Rechner verbunden, so daß sich eine hierarchische Struktur ergibt, die der üblichen Organisation eines größeren Unternehmens entspricht. Meistens ist dabei aber gefordert, daß die Arbeitsplatzrechner einer Abteilung und die Abteilungsrechner untereinander auch direkt zusammenarbeiten können (COL86, SCH87).

In einem solchen Netz sind die Leistungsanforderungen auf den verschiedenen Ebenen unterschiedlich. In der Zentrale sind Rechner hoher Leistung mit großen Datenspeichern (im Englischen als Hosts oder Mainframes bezeichnet) notwendig; auf Abteilungsebene sind die Anforderungen an Leistung und Speicher meist kleiner, so daß Minicomputer ausreichen; am Arbeitsplatz schließlich werden Microcomputer eingesetzt. Dabei steigen fortlaufend Leistung und Speicherfähigkeit in allen diesen Klassen.

Physische/logische  
Struktur

In einem Rechnernetz muß man grundsätzlich zwei Aspekte der Strukturierung unterscheiden: einmal gibt es eine physische Struktur, die die Hardware-Gegebenheiten des Netzes beschreibt; daneben hat das Netz aber eine logische Struktur, mit der es durch abstrakte Instanzen mit bestimmten Kommunikationsbeziehungen beschrieben wird.

Beide Strukturen müssen nicht unbedingt überall übereinstimmen: So kann z.B. ein Rechner an einen anderen, völlig gleichartigen Rechner wie eine Datenstation angeschlossen werden. Auch kann sich die logische Struktur auf verschiedenen Ebenen unterschiedlich darstellen: Ein Zentralrechner kann mit einem Abteilungsrechner durchaus peer-to-peer kommunizieren, obwohl beide in der Aufgabenverteilung eine hierarchische Beziehung haben.

Netzknoten,  
Endbenutzer

Das einfachste logische Bild eines Kommunikationssystems für verteilte Datenverarbeitung ist das von untereinander verbundenen Netzknoten, an die jeweils Endbenutzer angeschlossen sind (siehe Abb. 2.2-1). Dabei ist ein Netzknoten zunächst einfach eine (abstrakte) Instanz des netzumfangreichen Kommunikationssystems, die - in Zusammenarbeit mit den anderen Knoten des Systems - dessen Dienste erbringt. Im OSI-Referenzmodell wird ein Knoten zusammen mit seinen Endbenutzern als "System" gesehen; geregelt wird die Kommunikation zwischen Endbenutzern verschiedener Systeme.

Ein Endbenutzer kann ein Mensch an einer Datenstation oder ein Programm sein. Wie die Endbenutzer an "ihren" Knoten angeschlossen sind, wird nicht durch das OSI-Referenzmodell bestimmt und ist Sache der physischen Gegebenheiten: So ist z.B. ein Programm als Endbenutzer mit "seinem" Netzknoten über eine Programmschnittstelle gekoppelt, eine