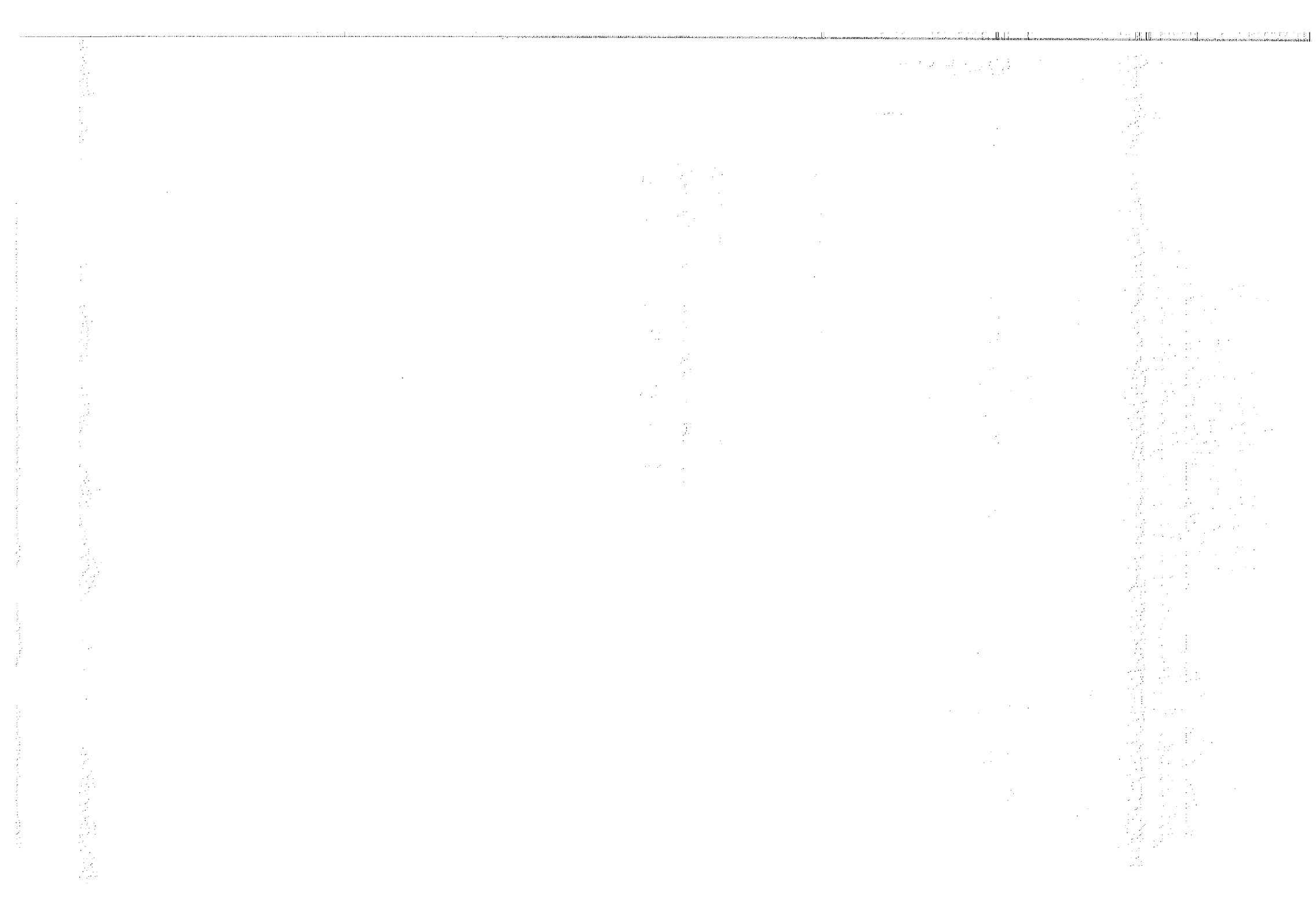
FORSCHUNGSZENTRUM JÜLICH GmbH

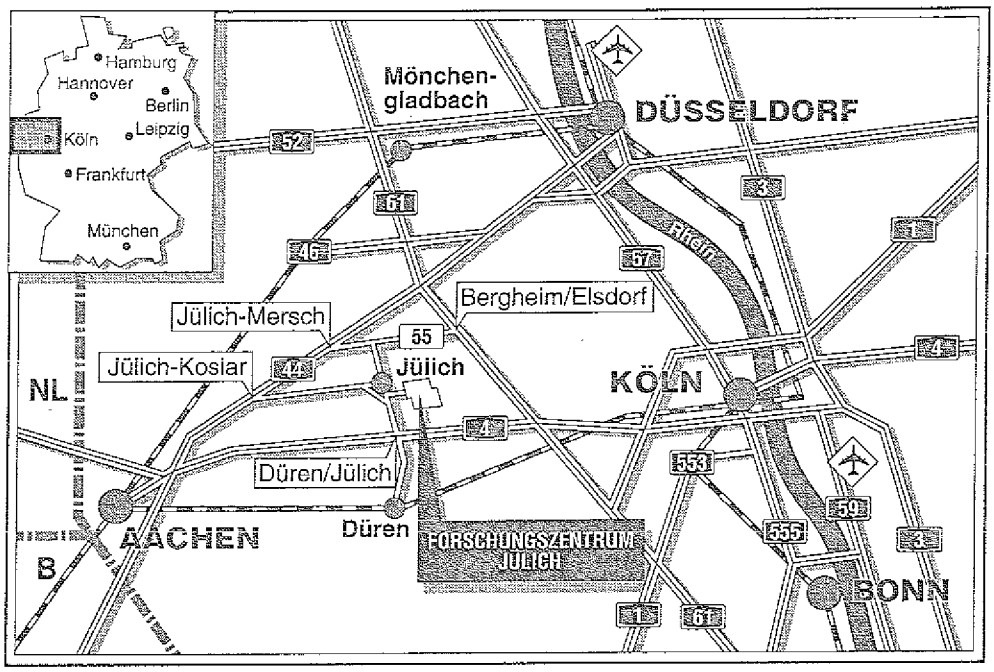
KFA

*Zentralinstitut für Angewandte Mathematik*

# Untersuchung über SNMP als Zwischenstufe zum OSI-orientierten Netzwerkmanagement

Mohammad Shahbaz






**Berichte des Forschungszentrums Jülich ; 2622**
ISSN 0366-0885
Zentralinstitut für Angewandte Mathematik Jül-2622

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek
Postfach 1913 · D-5170 Jülich · Bundesrepublik Deutschland
Telefon: 02461/61-6102 · Telefax: 02461/61-6103 · Telex: 833556-70 kfa d

# Untersuchung über SNMP als Zwischenstufe zum OSI-orientierten Netzwerkmanagement

Mohammad Shahbaz

# Inhaltsverzeichnis

# Verzeichnis der Abbildungen

# Verzeichnis der Tabellen

# 1 Einleitung

Im Zeitalter der verteilten Datenverarbeitung sind Datenverarbeitungs- und Kommunikationsressourcen von nahezu gleichrangiger Bedeutung. Das Netzwerk als Rückgrat der verteilten Datenkommunikation ermöglicht es, verschiedene Endsysteme, wie z.B. Mainframes, Minicomputer und Workstations, miteinander zu verbinden und kann selbst über Zwischensysteme, wie z.B. Bridges, Routers oder Gateways, mit anderen Netzwerken gekoppelt werden, so daß ein komplexes und geographisch weit verbreitetes Netz von Netzwerken *(internet)* entstehen kann. Daher wird unter dem Begriff 'Netzwerk' die Gesamtheit aller Komponenten verstanden, die erforderlich sind, um eine herstellerübergreifende Kommunikation zu realisieren. Die so ermöglichte netzüberschreitende Kommunikation ist für viele Organisationen, sei es Industrie, Dienstleistungsunternehmen oder öffentliche Verwaltung aber auch Wissenschaft und Forschung, die Grundlage zur Erfüllung ihrer Aufgaben. Darüber hinaus hängt der marktwirtschaftliche Erfolg vieler Unternehmen direkt von der Vernetzung komplexer moderner Kommunikationssysteme ab. So besteht die Notwendigkeit, Netze verschiedener Architektur zu verbinden oder Geräte unterschiedlicher Hersteller in das Netz einzubinden. Hieraus resultiert eine zunehmende Heterogenität der Netze, wobei unter einem heterogenen Netzwerk ein durch Netzwerkelemente verschiedener Hersteller zusammengesetztes Netzwerk zu verstehen ist. Der Übergang zu den heterogenen Netzwerken ist in der Regel notwendig, da kein Anbieter allein alle Bedürfnisse einer großen Organisation abdecken kann und spezialisierte Systeme effizienter als Universalsysteme sind. Aufgrund der Heterogenität heutiger Netzwerke ist das Netzwerkmanagement, d.h. das Überwachen, Steuern und Verwalten von Netzwerken, ein sehr komplexes Problem geworden [ScCR91].

Während die homogenen, herstellerspezifischen Netzwerke durch herstellereigene Management-Produkte in allgemeinen effizient überwacht und gesteuert werden, sind diese für offene Systeme und herstellerübergreifende Netzwerke nicht über die Produktgrenzen hinaus einsetzbar. Zur Lösung des Heterogenitätsproblems muß nach Standards gesucht werden, die ein umfassendes Netzwerkmanagement definieren, so daß die Heterogenität vor dem Anwender verborgen bleibt.

Die zwei für das Management heterogener Systeme in Frage kommenden Netzwerkmanagement-Standards sind das SNMP-Netzwerkmanagement und das OSI-Netzwerkmanagement. Hierbei muß erwähnt werden, daß der in dieser Arbeit verwendete Begriff SNMP-Netzwerkmanagement auf das Gesamtkonzept des Netzwerkmanagements hinweist, während SNMP *(Simple Network Management Protocol)* nur das Protokoll spezifiziert. So ist SNMP eine Komponente des SNMP-Netzwerkmanagements, obwohl es in der Literatur oft mit dem SNMP-Netzwerkmanagement gleichgesetzt wird. Das SNMP-Netzwerkmanagement ist durch IAB *(Internet Activities Board)* normiert worden und stellt deshalb keinen internationalen Standard dar; es hat aber schon eine weite Verbreitung gefunden und ist in verschiedenen Umgebungen implementiert worden. Die hohe Akzeptanz des SNMP-Netzwerkmanagements ist im wesentlichen auf seine praxisorientierte Entwicklungsstrategie, einfache Architektur und leichte Handhabung zurückzuführen. Das OSI-Netzwerkmanagement, dessen Standardisierungsprozeß noch nicht abgeschlossen ist, repräsentiert die Lösung der Internationalen Standardisierungsorganisation (ISO) für das Netzwerkmanagement-Problem und wird einerseits wegen

seiner internationalen Anerkennung und andererseits wegen zahlreicher Funktionalitäten und Mechanismen im Vergleich zu den existierenden Management-Lösungen als Netzwerkmanagement der Zukunft betrachtet [SaHu90, Jand91a]. Da der Standardisierungsprozeß noch nicht abgeschlossen ist, demzufolge also entsprechende auf dem OSI-Protokollsatz basierende Netze noch nicht verfügbar sind, fehlen auch die das OSI-Netzwerkmanagement unterstützenden Produkte.

Obwohl das SNMP-Netzwerkmanagement ursprünglich für einfache Netzwerkmanagement-Aufgaben entwickelt worden ist, ist es jedoch nicht zuletzt auch durch entsprechende Erweiterungen und insbesondere vor dem Hintergrund seiner weiten Verbreitung und hohen Akzeptanz von Bedeutung, diesen Netzwerkmanagement-Ansatz auf dem Migrationsweg zum OSI-orientierten Netzwerkmanagement zu untersuchen. Eine geeignete Zwischenlösung, die zugleich den hier getätigten Investitionen Rechnung trägt, ist auf jeden Fall notwendig, da zur Zeit keine akzeptablen OSI-Lösungen auf dem Markt existieren, obwohl auf Seite der Anwender ein entsprechender Bedarf besteht.

Die vorliegende Diplomarbeit stellt sich zur Behandlung der Problemstellung zwei Aufgaben. Erstens will sie die beiden Netzwerkmanagement-Ansätze auf der Basis der vorhandenen Standards beschreiben, wobei das OSI-Netzwerkmanagement auf den für diese Arbeit notwendigen Rahmen beschränkt bleiben soll. Zweitens soll sie aus dem Blickwinkel des SNMP-Netzwerkmanagements die beiden Netzwerkmanagement-Ansätze miteinander vergleichen.

# 2 OSI-Netzwerkmanagement

Die Strategie der Internationalen Standardisierungsorganisation (*International Standards Organization, ISO*) bei der Normierung des Netzwerkmanagements ähnelt der Vorgehensweise bei der Entwicklung von Normen zur Regelung der Datenübertragung, die als *Open Systems Interconnection - Basic Reference Model (OSI-BRM)* bezeichnet werden. Diese Methode bestand in der Schaffung eines Rahmens mit der Beschränkung der Beschreibung auf die Mechanismen der Datenübertragung. Im Bereich des Netzwerkmanagements entwickelte die ISO ein Konzept, das der Koordination und der Entwicklung der vorhandenen und zukünftigen OSI-Netzwerkmanagement-Standards dient. Die ISO-Standards für das Netzwerkmanagement haben die Aufgabe, ein herstellerunabhängiges Netzwerkmanagement zu realisieren. Die Standardisierung des OSI-Netzwerkmanagements im Rahmen offener Systeme umfaßt:

- allgemeine Netzwerkmanagement-Architektur und -Struktur,
- Funktionsbereiche,
- Syntax und Semantik von Management-Informationen,
- Dienste und Protokolle zum Austausch von Management-Informationen und
- die Management-Objekte als Gesamtheit aller OSI-relevanten Ressourcen eines Netzes.

Nach der kurzen Beschreibung der Internationalen Standardisierungsorganisation (ISO) und des ISO/OSI-BRM wird in diesem Kapitel das ISO/OSI-Netzwerkmanagement, auch OSI-Netzwerkmanagement genannt, erläutert. Dabei ist zu beachten, daß die Standardisierung des Netzwerkmanagements nicht abgeschlossen ist, und die Darstellung hier den aktuellen Zustand der Normen aufzeigt. Bei der Erstellung dieses Kapitels wurden hauptsächlich die Original-ISO-Dokumente bis Mitte 1991 benutzt.

## 2.1 Internationale Standardisierungsorganisation ISO

Die internationale Standardisierungsorganisation (ISO) hat die Aufgabe, auf der Basis von Vorlagen Kompromisse zu erzielen, die bei den Mitgliedern Übereinstimmung und bei den Benutzern Akzeptanz finden. Daher ist ISO auf die internationale Zusammenarbeit von Firmen und Institutionen angewiesen, wenn praktikable Normen in akzeptablen Zeitrahmen verabschiedet werden sollen. Die Aufgabe von ISO besteht also in der Schaffung internationaler Normen mit der Bezeichnung *International Standard (IS)*. ISO besitzt keine Macht zur Durchsetzung ihrer Standards. Die Hersteller werden diese Standards nur dann immer stärker anwenden, wenn sie sich davon geschäftlichen Erfolg versprechen. Die Motivation hierzu kann zum Beispiel die Forderung der großen Unternehmen und Behörden nach Einhaltung dieser Standards sein.

Das *Technical Committee 97 (TC 97)* ist für die Entwicklung von Standards im Kommunikationsbereich zuständig. Innerhalb des TC 97 existieren die vier *Sub-Committees (SCs)*, SC 6 *(Telecommunications)*, SC 16 *(Open Systems)*, SC 18 *(Text Preparation and Interchange)* und SC 19 *(Office Equipment and Supplies)*. Die eigentliche Sacharbeit wird in Arbeitsgruppen *(Working Groups, WGs)* geleistet [Conr89]. Für den Bereich der Informationstechnologie haben ISO und IEC *(International Electrotechnical Commission)* ein *Joint Technical Committee*, ISO/IEC JTC1, gegründet. Die Mitglieder

der ISO/IEC sind nationale Standardisierungsgremien, unter anderem ANSI *(American National Standards Institute)* , BSI *(British Standards Institution)* und DIN *(Deutsches Institut für Normung)*. Die ISO/IEC unterhält Verbindungen mit anderen multinationalen Standardisierungsorganisationen, wie IEEE *(Institute of Electrical and Electronic Engineers)* und ECMA *(European Computer Manufacturers' Association)*. Viele nationale Organisationen, staatlich oder nicht staatlich, nehmen an dieser Arbeit in Verbindung mit ISO und IEC teil [ISO7498-4]. Innerhalb von ISO/IEC sind die zwei *Sub-Committees* JTC1/SC6 und JTC1/SC21 für OSI zuständig, wobei JTC/SC6 für die Datenmobilität und JTC1/SC21 für die Informationsstruktur verantwortlich ist.

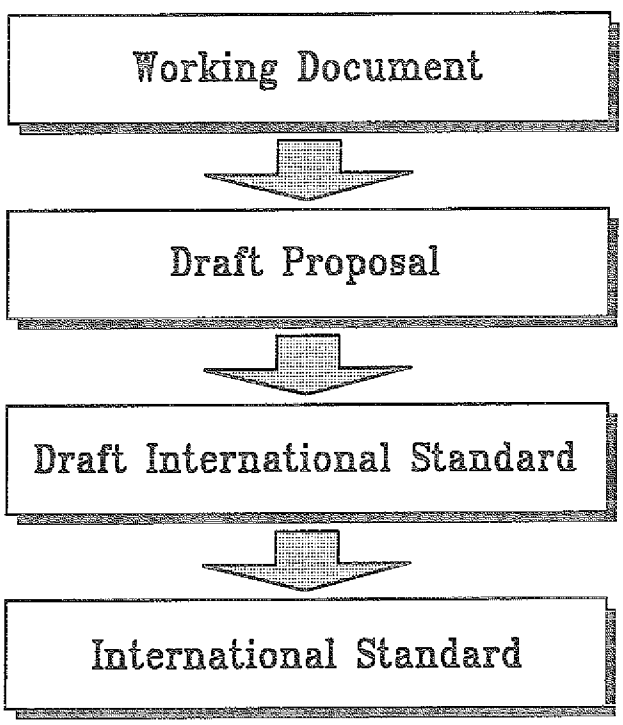



**Abb. 1. Standardisierungsstufen:** Dargestellt sind die bei einem Standardisierungsentwurf durchlaufenen Stadien.

Der Prozeß der Entstehung eines internationalen Standards ist langwierig und kompliziert. Von der Entstehung der Normen bis zur Annahme als *International Standard* werden verschiedene Stadien durchlaufen (Abb. 1):

1. Working Document (WD);
2. Draft Proposal (DP);
3. Draft International Standard (DIS);
4. International Standard (IS).

Das erste Stadium ist die Entstehung eines *Working Document (WD)* in den Arbeitsgruppen. Dem *Working Document* wird eine Normierungs-Nummer und ein *Sub-Committee* zugeordnet, das für die Entwicklung des Dokuments bis zum Endstadium der Normierung zuständig ist. Diese Nummer ist charakteristisch für dieses Dokument und

wird nicht geändert. Das WD ist häufigen Modifikationen unterworfen und gilt als recht instabil. Findet eine ausreichende Übereinstimmung über die Weiterentwicklung der WDs statt, so geht man zum zweiten Stadium über, d.h. das WD wird ein *Draft Proposal*. Das zweite Stadium befaßt sich mit der Erarbeitung von *Draft Proposals* in den *Working Groups* und *Sub-Committees* auf der Basis von *Working Documents*, die sich bis dahin als hinreichend stabil gezeigt haben und verfeinert wurden. Hier wird typisch eine Zeitspanne von sechs Monaten gesetzt, um erneut über die Weiterentwicklung des Dokuments zu entscheiden. Während dieser Zeitspanne finden Diskussionen statt, die mit Wahlen abschließen. Nach einer ausreichenden Zustimmung der Mitglieder, findet der Übergang zum dritten Stadium statt. Der Standard wird *Draft International Standard (DIS)* genannt. Dabei wird auch der Termin für die nächste Wahl gesetzt. Ein DIS ist, von der technischen Seite her gesehen, relativ stabil. Die einzigen Änderungen dürfen in der Regel keine technischen Probleme mehr betreffen, sondern nur noch typographische Fehler. Die durch dieses Komitee entwickelten DIS werden nach Billigung von mindestens 75% der ISO-Mitglieder als *International Standard (IS)* erklärt [Rose90a]. Die Syntax zur Kennzeichnung eines Dokuments lautet:

ISO/IEC << Zustand >> << Nummer >>.

Dabei bezeichnet << Zustand >> die Normierungsstufe, auf der sich ein Dokument befindet, wie WD oder DIS, und << Nummer >> die dem Dokument zugeordnete Normierungs-Nummer. Wenn ein Dokument als IS spezifiziert worden ist, wird gewöhnlich die obige Syntax ohne die Angabe des Zustands verwendet. Ein IS ist normalerweise für eine lange Zeit stabil. Werden aber Verbesserungen benötigt, so findet ein Nachtrag-Prozeß statt, der diese Aufgabe in drei folgenden Stufen erfüllt:

1. Proposed Draft Addendum (PDAD);
2. Draft Addendum (DAD);
3. Addendum (AD).

## 2.2 ISO/OSI-Referenzmodell

Für die bessere Darstellung des ISO/OSI-Referenzmodells ist das Verständnis des ihm zugrundeliegenden, strukturierten Schichtungs-Prinzips notwendig. Die Software und Hardware der Stationen, die auf Daten-Kommunikations-Netzwerken operieren, bestehen aus vielen Funktionen, die die Kommunikationsaktivitäten unterstützen. Die Anzahl und die Komplexität dieser Funktionen stellen zwei schwierige Aufgaben dar, mit denen die Netzwerkentwickler konfrontiert sind. Um diese Probleme in den Griff zu bekommen, sind viele Systeme mit einer gut geordneten und strukturierten Schichtung (*layering*) der Funktionen entworfen worden. Der diese Technik beschreibende Ausdruck heißt Schicht-Protokoll *(layered protocol)*. Ein gut entworfenes in Schichten organisiertes Protokoll-System kann folgendes zur Verfügung stellen [Blac91]:

- Logische Zerlegung eines komplexen Systems in kleinere, begreiflichere und unabhängig behandelbare Teile (Schichten);
- Standardisierte und vom Funktionalitätsumfang begrenzte Schnittstellen zwischen den in Schichten organisierten Funktionen;

- Funktions-Symmetrie auf jeder Schicht in einem System - in jeder Maschine führt dieselbe Schicht die gleiche(n) Funktion(en) aus;

Die Schicht-Protokolle, die den allgemeinen Konventionen der ISO unterworfen sind, erlauben verschiedenen Herstellersystemen ohne Änderung der Schichtenlogik *offen* miteinander Informationen auszutauschen. Deshalb sollen die sogenannten geschlossenen *(closed)* Systeme, die bis jetzt nur begrenzt, bilateral Informationen austauschen können, durch ISO/OSI konforme Kommunikations-Software abgelöst werden. Unter *Open Systems Interconnection (OSI)* ist also die Kommunikation offener Systeme zu verstehen.

Der Ursprung der Zerlegung der Funktionen und ihre Zuordnung zu den unterschiedlichen Schichten liegt bei den verschiedenen Konzepten, die im allgemeinen strukturierte Techniken genannt werden. Diese Ideen werden beim Entwurf von Hardware- und Software-Systemen, die eine genaue Definition der Schnittstellen beinhalten, immer stärker benutzt. Die Systeme umfassen dabei verschiedene Module, die jeweils eine Funktion oder einige eng verwandte Funktionen ausführen. Diese Techniken können auch eine lose Verbindung *(loose coupling)* zur Verfügung stellen, wobei eine Änderung eines Moduls in einem System keine Wirkung auf irgendeine andere Komponente hervorruft. Werden zusätzlich die Prinzipien der ***atomaren Aktionen*** berücksichtigt, erlauben die Protokolle in einem in Schichten organisierten Netzwerk eine Wechselwirkung zwischen den funktionell gleichen Schichten in verschiedenen Systemen, ohne die anderen Schichten zu beeinflussen. Dieses Konzept hilft bei der Verteilung der Funktionen zu den Schichten. Die Definition einer atomaren Aktion wird verschieden aber eineindeutig gehandhabt[1]:

1. „Eine Aktion ist atomar, wenn sie keine Kenntnis von der Existenz anderer aktiver Prozesse hat, und andere Prozesse nichts vom Prozeß X wissen, während der Prozeß X die Aktion ausführt. Dies besagt, daß Prozesse, wie die Schichten, unabhängig operieren sollen;

2. Eine Aktion ist atomar, wenn der die Aktion ausführende Prozeß nicht mit anderen Prozessen kommuniziert, während die Aktion durchgeführt wird. Dies bedeutet, daß die Abwicklung der Kommunikation zwischen einem Prozeß (einer Schicht) und einem anderen Prozeß (einer anderen Schicht) auf den Zeitabschnitt eingeschränkt ist, in dem die Schichten Daten und Zustandsinformationen austauschen, d.h. nur während der Ausführung des Schichtprotokolls;

3. Eine Aktion ist atomar, wenn der die Aktion ausführende Prozeß keine Änderungen, außer den durch sich selbst hervorgerufenen Änderungen, entdecken kann, und wenn er seine Zustandsänderungen nicht offenbart, bis die Aktion ausgeführt ist. Anderes ausgedrückt, ein laufender Prozeß soll nicht unterbrochen werden, wenn die Unterbrechung eine Änderung des Prozeßergebnisses zur Folge hat".

Das ***OSI-Basis-Referenzmodell*** *(OSI-BRM)* [ISO7498], das in Schichten organisiert ist, wurde entworfen, um den Kommunikationsvorgang zu regeln. Der Kommunikations-

---

1 siehe Seite 6 in [Blac91]

vorgang wird in sieben Schichten zerlegt. In jeder Schicht sind Instanzen *(entities)* vorhanden, die die Schicht-bezogenen Aufgaben erfüllen. Die Anordnung ist hierarchisch aufgebaut. Bei der vertikalen Kommunikation kann jede Instanz einer Schicht N nur das Dienstangebot der direkt darunterliegenden Schicht N-1 in Anspruch nehmen und ihre eigenen Dienste nur einer Instanz der direkt darüberliegenden Schicht N+1 zur Verfügung stellen (Abb. 3) [JaAg90].

Die Dienste der Instanzen einer Schicht werden über Schnittstellen, die Dienstzugangspunkte *(Service Access Points, SAPs)* genannt werden, der darüberliegenden Schicht angeboten. Horizontal kann eine Instanz einer Schicht mit einer Partnerinstanz *(peer entity)* kommunizieren. Die Partnerinstanz befindet sich auf der funktionell gleichen Schicht zum Beispiel in einer anderen Maschine. Bei der Kommunikation zwischen Partnerinstanzen, die durch Protokolle geregelt wird, werden Daten und Steuerinformationen in Form von Protokoll-Dateneinheiten *(Protocol Data Units, PDUs)* unter Inanspruchnahme der Dienste der darunterliegenden Schichten ausgetauscht. Unter einem Protokoll einer Schicht versteht man einen Satz von Regeln, die den Austausch von Informationen zwischen den Partnerinstanzen festlegten. Die PDUs werden durch den Transport von einer Instanz der Schicht N+1 zur darunterliegenden Schicht N um einen Nachrichtenkopf *(header)* ergänzt, der die Protokoll-Steuerungs-Information *(protocol control information, PCI)* enthält. Die Erweiterung der Anwender-Daten um Nachrichtenköpfe wird *encapsulation* genannt. In der anderen Maschine entfernt jede Instanz den für sie bestimmten Nachrichtenkopf. Nach der Auswertung übergibt sie den Rest der PDU der nächsten Instanz in der darüberliegenden Schicht. Das Entfernen der Protokoll-Steuerungs-Information wird entsprechend *decapsulation* genannt [Stal89a].

Die verschiedenen Schichten in der OSI liefern den darüberliegenden Schichten Dienste (Ausnahme: die Anwendungsschicht) und nehmen die Dienste der darunterliegenden Schichten in Anspruch (Ausnahme: die Bitübertragungsschicht). Zur Verdeutlichung des Mechanismus, wie eine OSI-Schicht als Dienst-Anbieter fungiert, sollen zwei Anwender A und B betrachtet werden, die die Dienste einer OSI-Schicht nutzen. Die Anwender kommunizieren mit dem Dienst-Anbieter durch einen SAP, der als eine Adresse zur Identifikation aufzufassen ist.

Die ganze Prozedur zur Bereitstellung eines Dienstes wird nach OSI-Vorstellung in vier Teil-Prozeduren unterteilt, die Dienstelemente *(primitives)* genannt werden:

1. Die Aufforderung *(request)* ist ein beim Dienst-Benutzer angesiedeltes Dienstelement, das zum Aufruf einer Funktion dient;

2. Der Hinweis *(indication)* ist ein Dienstelement des Dienst-Anbieters, das zum Aufruf einer Funktion oder zum Hinweis, daß eine Funktion am SAP aufgerufen worden ist, benutzt wird;

3. Die Beantwortung *(response)* ist ein beim Dienst-Benutzer angesiedeltes Dienstelement, das eine vorher durch den Hinweis-Dienstelement aufgerufene Funktion am SAP vollendet;

4. Die Bestätigung *(confirm)* ist wieder ein Dienstelement des Dienst-Anbieters, das eine vorher durch Anforderung aufgerufene Funktion am SAP vervollständigt.

Ein Dienstelement wird mit einem sehr spezifischen Format kodiert [Blac91]. Um zum Beispiel eine Verbindung zu einem entfernten Rechner herzustellen, könnte das folgende Dienstelement unter der Annahme, daß das Netzwerk mit dem Dienst-Benutzer durch die Dienstelemente kommuniziert, verwendet werden:

```
N-CONNECT request [Adressat, Adressant, QOS-Parameters, Anwenderdaten].
```

Die Dienstelement-Parameter sind die zur Identifikation dienenden Adressen des Empfängers und des Absenders. Die Dienstgüte-Parameter *(quality-of-service parameters, QOS parameters)* informieren den Dienst-Anbieter über den Typ des aufzurufenden Dienstes, wie z.B. beschleunigte Lieferung, während die Anwenderdaten-Parameter die tatsächlichen, zum Empfänger zu sendenden Daten darstellen.

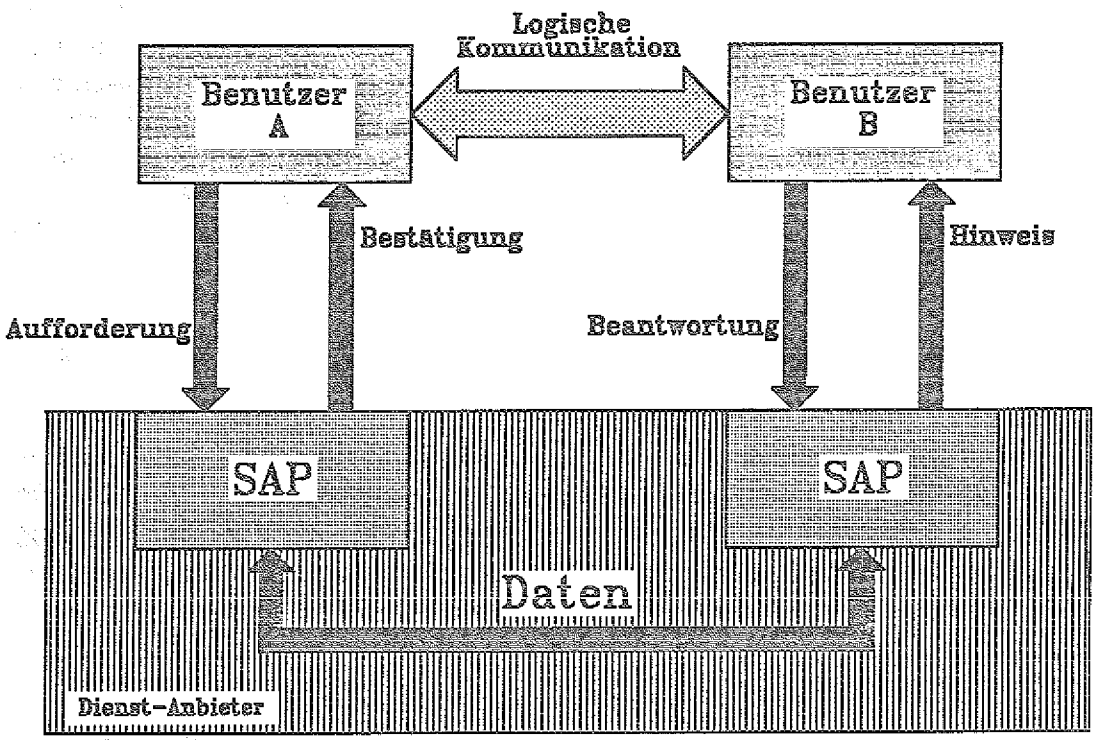



**Abb. 2. Die Dienstelemente in OSI:** Beschrieben werden die vier im OSI definierten Dienstelemente. Jeder Dienst wird durch den Austausch dieser Dienstelemente zwischen dem Dienst-Anbieter, der eine Schicht sein kann, und den Dienst-Benutzern zur Verfügung gestellt.

Ein Dienstelement wird von der Schicht benutzt, um die Dienstinstanzen *(service entities)* aufzurufen und Nachrichtenköpfe zu erzeugen, die dann in der entfernten Station durch die Partner-Schicht bewertet werden. Die Dienstelemente werden auch von den benachbarten Schichten benutzt, um Nachrichtenköpfe zu schaffen, die im entfernten System durch die Partner-Schichten benutzt werden.

Die vertikale Kommunikation, d.h. die Kommunikation zwischen Instanzen benachbarter Schichten, wurde nicht standardisiert. Dagegen ist die horizontale Kommunika-

tion, d.h. die zwischen Partnerinstanzen, Gegenstand der Normung gewesen [Conr89]. Das Referenzmodell befaßt sich lediglich mit dem Satz der Aufgaben, die die jeweilige Schicht erfüllen soll.

Die ISO hat die Normen für alle Schichten gebilligt. Diese Normen sind unabhängig vom ISO/OSI-Referenzmodell veröffentlicht worden und deshalb als außerhalb des Referenzmodells stehend zu betrachten.

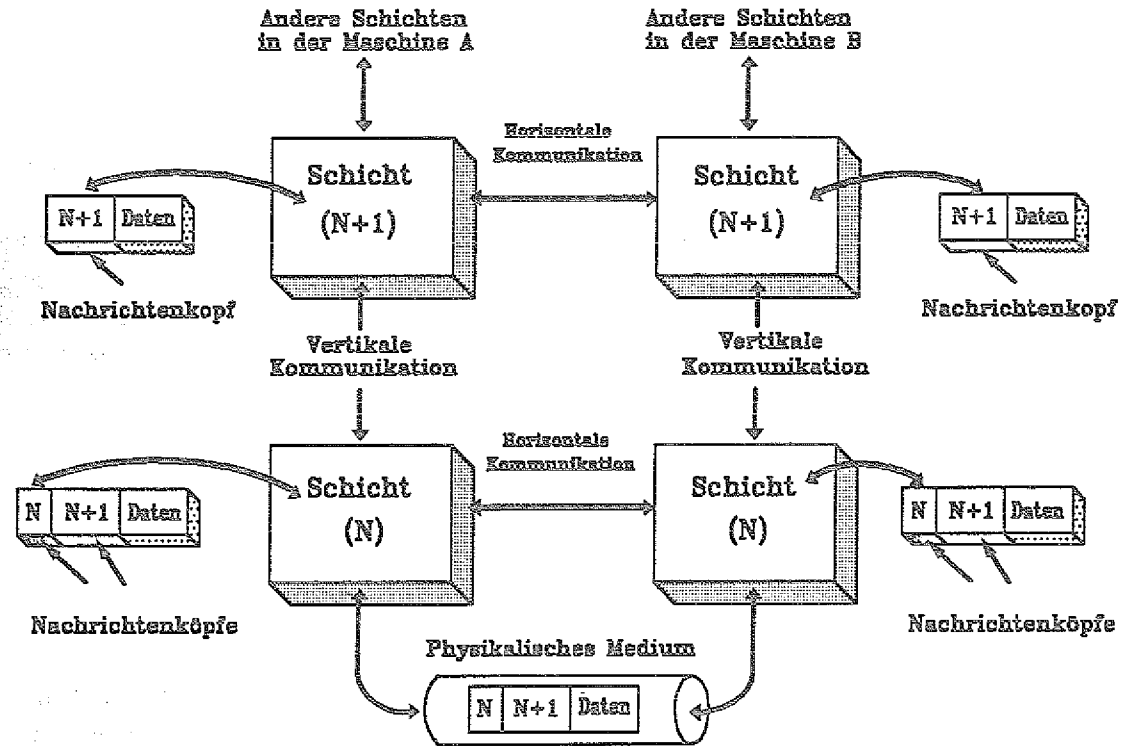



**Abb. 3. Schicht-Kommunikation [Biac91]:** Jede Instanz in Schicht N + 1 in der Maschine A kommuniziert mit ihrer Partnerinstanz in der Maschine B durch Inanspruchnahme der Dienste der darunterliegenden Schichten. Die PDUs werden in jeder Schicht mit einem Nachrichtenkopf ergänzt und der unteren Schicht weitergegeben. Nach der Übertragung werden die Nachrichtenköpfe jeweils durch die Schichten der Maschine B entfernt, und nach der Auswertung wird der Rest der darüberliegenden Schicht weitergegeben.

Die sieben Schichten des ISO/OSI-Referenzmodells, das die Aufgabe hat, offene Systeme miteinander zu verbinden, werden hier kurz erläutert. Unter offenen Systemen versteht man solche Systeme, die für die Kommunikation mit anderen Systemen offen sind [Tane90].

**Die Bitübertragungsschicht** *(physical layer)*, also das Übertragungsmedium, hat die Aufgabe, Bits über einen Kommunikationskanal zu übertragen, d.h. der Bitstrom wird aufgenommen und übertragen, ohne auf seine Struktur und seine Bedeutung zu achten. Diese Schicht befaßt sich zum Beispiel mit den Fragen, wie ein Bit elektrisch dargestellt und mit welcher Geschwindigkeit es übertragen werden soll, oder wieviele Anschlüsse (pins) eine Endsystemverbindung besitzt und wofür sie vorgesehen sind (vgl. Abb. 4).

**Die Sicherungsschicht** *(data link layer)* dient der gesicherten Übertragung von Informationseinheiten und der Adressierung der am Übertragungsmedium angeschlossenen Stationen. Die Informationseinheiten sind Datenrahmen *(data frames)*, die durch blockweise Aufteilung der Eingangsdaten und Erweiterung um einen Fehlercode zur Fehlererkennung und -behebung entstehen. Eine weitere Aufgabe dieser Schicht ist die Flußkontrolle. Die Fehlerbehandlung dieser Schicht umfaßt die Fehler, die bei der Übertragung entstanden sind. Bei lokalen Netzen wird die Sicherungsschicht in die Teilschichten *Media Access Control (MAC)*, die den Zugriff zum Übertragungsmedium steuert, und die darüberliegende Teilschicht *Logical Link Control (LLC)*, die die anderen vom Medienzugriff unabhängigen Dienste der Sicherungsschicht übernimmt, aufgeteilt [Stal89b].

**Die Vermittlungsschicht** *(network layer)* beschäftigt sich vor allem mit der Wegewahl *(routing)* und der Errichtung virtueller Pfade zwischen den Stationen des Netzes. Die Fehlerbehandlung und die Flußkontrolle zwischen den Endsystemen einer Verbindung erweitern das Aufgabengebiet dieser Schicht. Die Flußkontrolle innerhalb dieser Schicht dient der Vermeidung der Überlastung der Endpunkte einer virtuellen Verbindung. Die Flußkontrolle ist hier trotz der Flußkontrolle der Sicherungsschicht notwendig, da über eine Teilstrecke mehrere virtuelle Verbindungen mit verschiedenen Ausgangs- und Endpunkten hergestellt werden können. Die Fehlerbehandlung auf dieser Schicht bezieht sich auf die Fehler, die in Verbindung mit dem Routing auftreten. Die Aspekte der Flußkontrolle und der Fehlerbehandlung zwischen den Endsystemen einer Verbindung (nicht zwischen den Anwenderprozessen) werden in dieser Schicht behandelt. Die Vermittlungsschicht berücksichtigt also die Ende-zu-Ende-Aspekte der physikalischen Verbindung zwischen den Endknoten der Verbindung.

**Die Transportschicht** *(transport layer)* unterstützt die Verbindungen zwischen den Prozessen in den Endsystemen. Die Transportschicht hat die Aufgabe, Transportverbindungen aufzubauen. Beim Aufbau der Verbindung wird über die Dienstmerkmale zwischen den Partnern verhandelt, wie zum Beispiel die Forderung nach einem bestimmten Durchsatz. Zu den Aufgaben der Transportschicht gehört auch die Übernahme der Daten von der darüberliegenden Schicht, der Sitzungsschicht, und, wenn nötig, die Aufteilung *(segmenting)* oder das Zusammenfassen *(blocking)* der Daten in sinnvoll behandelbare Einheiten bzw. die entsprechenden Umkehroperationen auf der Empfangsseite *(reassembly / deblocking)*. Eine andere Aufgabe ist die Flußkontrolle zwischen den kommunizierenden Prozessen. Die Transportschicht befaßt sich mit den Ende-zu-Ende-Aspekten einer Verbindung zwischen Prozessen [Conr89].

**Die Sitzungsschicht** *(session layer)* liefert einen Mechanismus für die Steuerung eines Dialogs zwischen zwei Anwendungen. Die Sitzungsschicht stellt zumindest ein Mittel zur Verfügung, das den beiden Anwendungsprozessen erlaubt, eine Verbindung zu errichten und zu benutzen. Die Errichtung und die Benutzung einer Verbindung wird Sitzung *(session)* genannt. Eine Kommunikation kann logisch in die Phasen Verbindungsaufbau, Datentransfer und Verbindungsabbau unterteilt werden. Die Station, die eine Verbindung aufbauen will, sendet eine Verbindungsanforderung *(S-Connect-Request)* an die Partnerstation aus. Die gerufene Partnerstation schickt eine Antwort *(S-Connect-Response)* zurück. Empfängt die rufende Station eine Einwilligung auf die Anfrage, so sendet sie eine Bestätigung *(S-Connect-Confirm)* an die rufende Station zurück. Bei

dieser Kommunikation zwischen den Partnerstationen werden Parameter ausgetauscht, die die Funktionalität der aufzubauenden Verbindung beschreiben. Dazu gehören z.B. die Flußkontroll-Parameter, die Größe der Puffer, die auf der Sitzungsschicht bereitzustellen, sind und die Kommunikationsarten. Mögliche Kommunikationsarten sind die Übertragung in eine Richtung *(simplex)*, nicht gleichzeitig in beide Richtungen *(half-duplex)* und gleichzeitig in beide Richtungen *(full-duplex)* [Stal89b]. Diese Parameter werden ausgehandelt und erlauben hierdurch Partnerstationen unterschiedlicher Funktionalität zusammenzuarbeiten. Die Sitzungsschicht kann einen Mechanismus liefern, der es erlaubt, im Falle eines Netzwerkabsturzes, nicht alle Daten seit dem Verbindungsaufbau neu transferieren zu müssen, sondern nur die Daten, die nach dem letzten Fixpunkt *(checkpoint)* im Datenfluß, der auch Synchronisationspunkt *(synchronization point)* genannt wird, gesendet wurden.

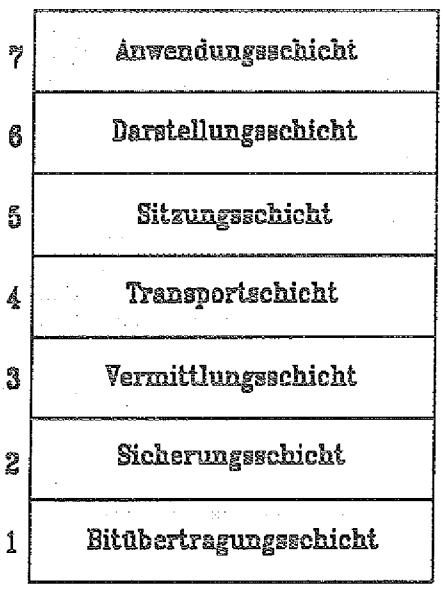



**Abb. 4. OSI-Referenzmodell**

**Die Darstellungsschicht** *(presentation layer)* befaßt sich mit der Syntax der zwischen den Anwendungsinstanzen ausgetauschten Daten und der Aufgabe, die Unterschiede bei der Informationsdarstellung in den kommunizierenden Systemen zu überbrücken. Die Datentypen, -werte und -strukturen einer Anwendung werden nach dem ISO-Konzept abstrakt beschrieben. Dazu wird die Beschreibungssprache ASN.1 *(Abstract Syntax Notation One)* verwendet. Die Darstellungsschicht setzt die lokale Syntax in die vorher vereinbarte Transfersyntax um. Im Zielsystem findet dann eine Umsetzung der Transfersyntax in die im Zielsystem geltende lokale Syntax statt. Zu weiteren Aufgaben in der Darstellungsschicht zählen die Datenkompression und die Kryptographie.

**Die Anwendungsschicht** *(application layer)* stellt ein Mittel zur Verfügung, das den Anwendungsprozessen erlaubt, auf die OSI-Umgebung zuzugreifen. Diese Schicht beinhaltet Managementfunktionen und auch allgemein nützliche Mechanismen zur Unterstützung verteilter Anwendungen. Im Gegensatz zur Darstellungsschicht befaßt sich die Anwendungsschicht mit der Semantik der Daten. Diese Schicht umfaßt Dienstelemente,

die die Anwendungsprozesse, wie *job management, file transfer, electronic mail* und *virtual terminal* unterstützen. Auch die Directory-Dienste werden durch diese Schicht bereitgestellt.

## 2.3 OSI-Netzwerkmanagement-Terminologie

Die ISO/IEC-Standards beinhalten unter anderem die Definition der im jeweiligen Standard verwendeten neuen Begriffe. Die wichtigsten Begriffe, die das OSI-Netzwerkmanagement betreffen, sind [ISO7498, ISO7498-4, ISO10040]:

**Real system** Ein reales System ist ein Satz von einem oder mehreren Computern, der dazugehörigen Software, der Peripherie, den Terminals, den menschlichen Operatoren, den physikalischen Prozessen, den Hilfsmitteln zur Informationsübertragung usw. und stellt ein selbstständiges Gebilde zur Informationsverarbeitung oder -übertragung dar.

**Real open system** Ein reales offenes System ist ein reales System, das sich in der Kommunikation mit anderen realen Systemen nach OSI-Empfehlungen richtet.

**Open system** Ein Offenes System ist die Darstellung der OSI-betreffenden Aspekte eines realen offenen Systems im OSI-Referenzmodell.

**(N)-subsystem** Ein (N)-Subsystem ist ein Element der hierarchischen Einteilung eines offenen Systems, das eine direkte Wechselwirkung nur mit den Elementen der nächst höheren oder nächst niedrigeren Einteilung desselben offenen Systems haben.

**(N)-layer** Eine (N)-Schicht ist eine Unterteilung der OSI-Architektur, die sich aus den Subsystemen des selben Rangs zusammensetzt.

**(N)-entity** Eine (N)-Instanz ist ein aktives Element innerhalb eines (N)-Subsystems.

**(N)-function** Eine (N)-Funktion ist ein Teil der Aktivität der (N)-Instanzen.

**(N)-protocol** Ein (N)-Protokoll ist ein Satz von Regeln und Formaten (Semantik und Syntax), die das Kommunikations-Verhalten der (N)-Instanzen bei der Ausführung der (N)-Funktionen bestimmen.

**(N)-connection** Eine (N)-Verbindung ist eine Verbindung, die von der (N)-Schicht zwischen zwei oder mehreren (N + 1)-Instanzen für die Übertragung von Daten errichtet wird.

**(N)-protocol-control-information**
Eine (N)-Protokoll-Steuerungs-Information ist die Information, die unter der Benutzung einer (N-1)-Verbindung zwi-

schen den (N)-Instanzen ausgetauscht wird, um ihre Verbindung zu koordinieren.

**(N)-user-data** (N)-Anwender-Daten sind die Daten, die zwischen den (N)-Instanzen im Interesse der (N+1)-Instanzen, denen die (N)-Instanzen Dienste bereitstellen, übertragen werden.

**(N)-protocol-data-unit** Eine (N)-Protokoll-Dateneinheit ist eine Einheit von Daten, die in einem (N)-Protokoll spezifiziert wird, und aus der (N)-Protokoll-Steuerungs-Information und möglicherweise (N)-Anwender-Daten besteht.

**OSI resources** Die OSI-Ressourcen sind die Datenverarbeitungs- und Datenkommunikations-Ressourcen, die OSI betreffen.

**Systems management** Das System-Management umfaßt die Funktionen in der Anwendungsschicht, die mit dem Management der verschiedenen OSI-Ressourcen und ihren Status über allen Schichten der OSI-Architektur verbunden sind.

**Systems management application-entity**
Die System-Management-Anwendungsinstanz ist eine Anwendungsinstanz zum Zweck der System-Management-Kommunikation und Ausführung der System-Management-Funktionen.

**OSI management** Das OSI-Netzwerkmanagement umfaßt die Software- und Hardware-Einrichtungen zur Steuerung, Koordination und Überwachung der Ressourcen, die es erlauben, daß eine Kommunikation in der OSI-Umgebung stattfindet.

**(N)-layer-operation** Die (N)-Schicht-Operation befaßt sich mit der Überwachung und Steuerung einer einzelnen Kommunikationsinstanz auf der (N)-Schicht.

**Managed object** Das verwaltete Objekt ist die OSI-Netzwerkmanagement-Sicht von einer Ressource innerhalb der OSI-Umgebung, die durch OSI-Management-Protokolle verwaltet werden kann.

**Management information base**
Die Management-Informationsbasis (MIB) ist ein konzeptioneller Ort zur Aufbewahrung der Management-Information innerhalb eines offenen Systems (siehe auch „2.5.4 Management-Informationsbasis“ auf Seite 38).

**Application process** Der Anwendungsprozeß (AP) ist eine Komponente innerhalb eines realen offenen Systems und stellt die abstrakte Repräsentation der Elemente eines offenen Systems dar, die eine bestimmte Anwendung ausführen.

**Application entity** Die Anwendungs-Instanz (AE) repräsentiert den Anwendungsprozeß innerhalb der OSI und stellt einen Satz von OSI-Kommunikations-Fähigkeiten für den Anwendungsprozeß (AP) bereit.

**Protocol machine** Die Protokoll-Maschine ist eine Maschine mit begrenzten Zuständen *(finite state machine, FSM)*, die ein bestimmtes Protokoll implementiert [Rose89]. Wenn als Eingabe nach einem bestimmten Dienst oder einer Netzwerkaktivität in einem bestimmten Zustand verlangt wird, erzeugt sie als Ausgabe z.B. einen Dienst-Hinweis oder eine Netzwerkaktivität und geht möglicherweise in einen anderen Zustand über [Rose90a].

**Application protocol data unit** Die Anwendungs-Protokoll-Dateneinheit (APDU) ist ein Datenobjekt, das durch Protokollmaschinen ausgetauscht wird, und umfaßt in der Regel die Protokoll-Steuerungs-Informationen und die Anwender-Daten.

**Application service element** Ein Anwendungsdienstelement (ASE) ist ein Teil der Anwendungsinstanz, die einen für einen speziellen Zweck definierten OSI-Dienst liefert. Es erlaubt den Anwendungsinstanzen (AEs) der gleichen Art durch Verwendung von Anwendungs-Protokoll-Dateneinheiten (APDUs) zusammenzuarbeiten.

## 2.4 ISO/OSI-Netzwerkmanagement-Architektur

Das Management, das die Kontrolle und die Überwachung von Ressourcen in offenen Systemen zur Aufgabe hat, beinhaltet verschiedene Aspekte [Rose89]. Die Ressourcen innerhalb offener Systeme können der Datenspeicherung, -verarbeitung und -übertragung dienen. Die Ressourcen, die bei der Übertragung eine Rolle spielen und die Kommunikation zwischen den beteiligten Systemen durch Informationstransfer ermöglichen, sind der OSI-Netzwerkmanagement-Standardisierung unterworfen (vgl. Abb. 6).

Die innerhalb der ISO für die Standardisierung des Netzwerkmanagements zuständigen Arbeitsgruppen haben beim Standardisierungsprozeß einen Rahmen entworfen, der der Verwaltung der einzelnen OSI-Schichten, der OSI-Systeme selbst sowie vollständiger Netze dienen soll. Die beiden ISO-Dokumente Management-Rahmenwerk *(management framework)* [ISO7498-4] und System-Management-Überblick *(systems management overview)* [ISO10040] liefern einen allgemeinen Überblick von den OSI-Netzwerkmanagement-Standards und stellen die Architektur und das Modell des OSI-Netzwerkmanagements dar.

Die OSI-Netzwerkmanagement-Standards beinhalten erstens die Dienste und die Protokolle, die zur Übertragung von Management-Informationen zwischen den offenen Sy-

stemen benutzt werden, und zweitens die abstrakte Syntax und Semantik der durch Management-Protokolle übertragenen Informationen (vgl. Abb. 5).

### 2.4.1 Management-Rahmenwerk

Das Management-Rahmenwerk entstand, um die Entwicklung der vorhandenen und zukünftigen OSI-Netzwerkmanagement-Normen zu koordinieren und diesen Normen als Referenz zu dienen. Das Rahmenwerk definiert die Terminologie und Konzepte, die beim OSI-Netzwerkmanagement verwendet werden und beschreibt seine Struktur zusammen mit einem Überblick von Zielen und Diensten. Schließlich werden die OSI-Netzwerkmanagement-Aktivitäten dargestellt [ISO7498-4].

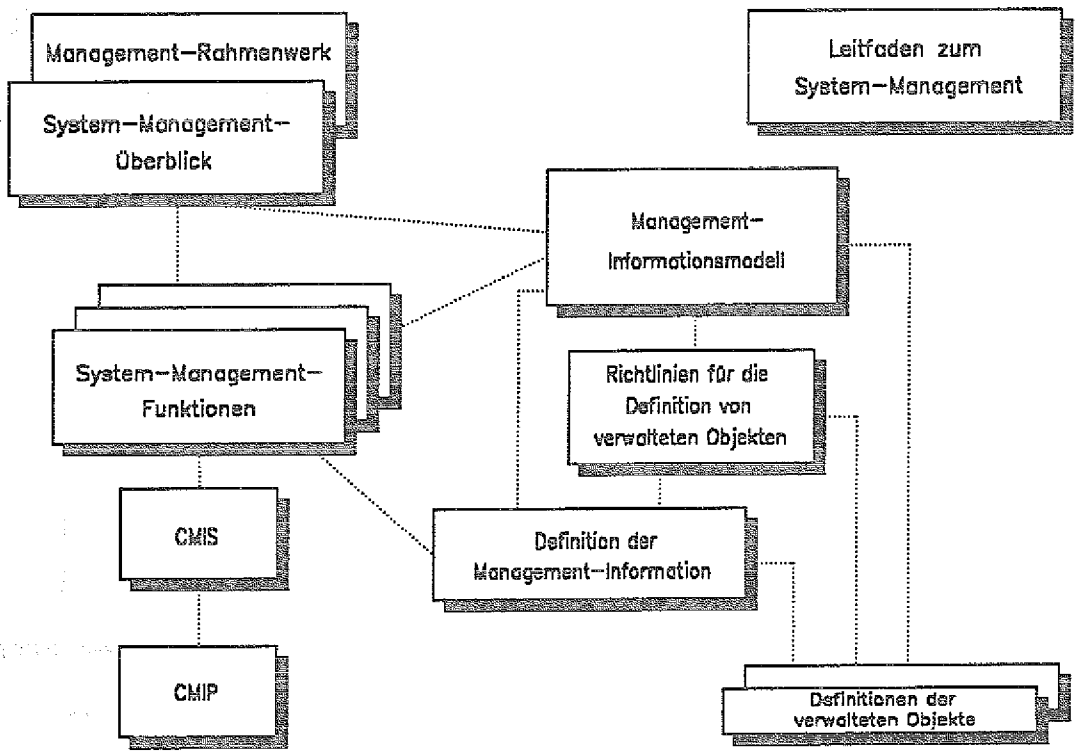



**Abb. 5. OSI-Netzwerkmanagement-Standards:** Dargestellt sind die Standards des OSI-Netzwerkmanagements und ihre Beziehung zueinander. Die Gesamtheit dieser Standards wird in dieser Arbeit als OSI-Netzwerkmanagement-Rahmenwerk bezeichnet.

### 2.4.2 Konzepte im OSI-Netzwerkmanagement

Die Konzepte im OSI-Netzwerkmanagement befassen sich mit vier grundlegenden Fragen:

1. Was verwaltet werden soll,
2. wer verwalten soll,
3. wie verwaltet werden soll und
4. welche Aufgaben erfüllt werden sollen.

Das Konzept des verwalteten Objekts beschreibt, wie das OSI-Netzwerkmanagement die zu verwaltenden Objekte sieht, und wie sie zu Managementzwecken repräsentiert werden. Das Manager-Agent-Konzept ist die Betrachtungsweise des OSI-Netzwerkmanagements, die die Rollen des Verwalters und der Verwalteten in der OSI-Umgebung hervorhebt. Die Art der Verwaltung wird durch die Kommunikation der offenen Systeme miteinander verdeutlicht. Die Anforderungen der Anwender definieren, welche Aufgaben erfüllt werden müssen. Sie werden in Funktionsbereiche kategorisiert, die im Abschnitt „2.5.1.3 Funktions-Aspekte" auf Seite 26 beschrieben werden.

#### 2.4.2.1 Forderungen der OSI-Netzwerkmanagement-Benutzer

Das OSI-Netzwerkmanagement unterstützt Benutzer, wobei unter Benutzer die Systemverwalter verstanden werden sollen. Diese Unterstützung kann in Form von Aktivitäten erfolgen, die einem Netz-Manager bei der Planung, Organisation und Überwachung von Netzen sowie der Abrechnung benutzter Kommunikations-Ressourcen dienen, oder solche Hilfsmittel, die ein vorhersagbares Kommunikationsverhalten sicherstellen. Die Unterstützung kann auch die Fähigkeit sein, auf veränderte Anforderungen reagieren zu können. Schließlich umfaßt sie diejenigen Werkzeuge, die für den Schutz der Informationen und für die Identifikation der Informationsquelle oder -senke der übertragenen Informationen dienen.

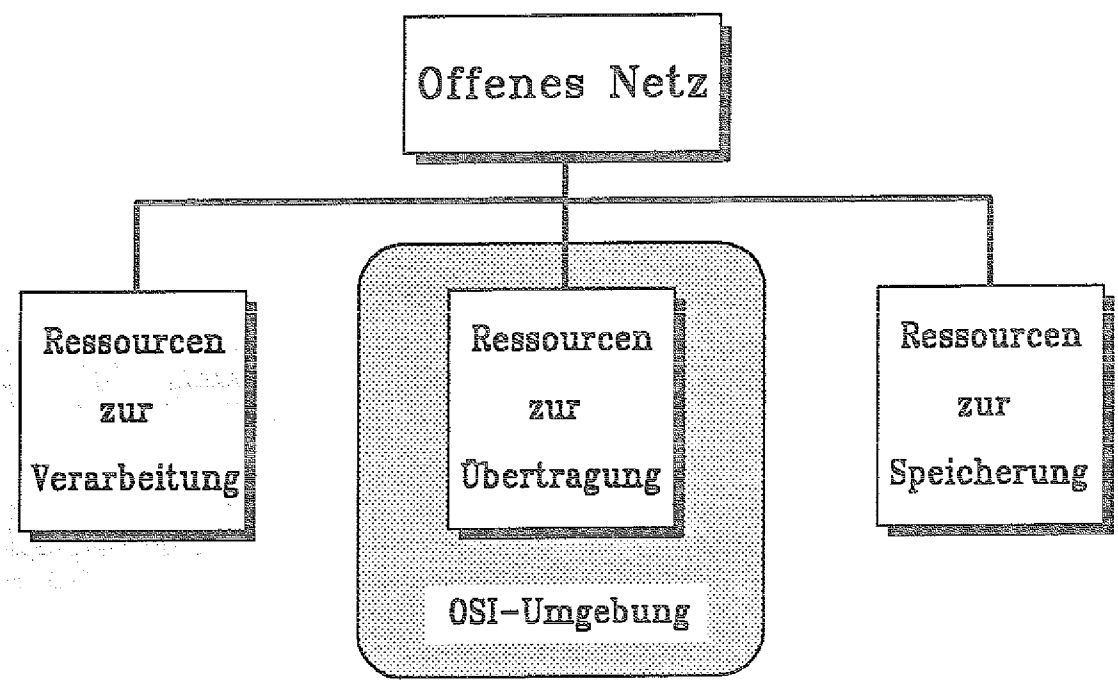



**Abb. 6. OSI-Umgebung:** In einem offenen Netz existieren Ressourcen zur Verarbeitung, Übertragung und Speicherung von Daten. Nur die Ressourcen zur Übertragung der Daten werden im Rahmen der OSI-Netzwerkmanagement-Standardisierung berücksichtigt [Rose89].

Die Komplexität der Management-Werkzeuge *(management tools)*, die diese Unterstützung ermöglichen, hängt von den Forderungen der Benutzer ab. Die Management-

Werkzeuge können lokal oder zwischen verschiedenen offenen Systemen operieren. Die Benutzer-Schnittstelle wird vom OSI-Netzwerkmanagement nicht festgelegt.

#### 2.4.2.2 OSI-Netzwerkmanagement-Umgebung

Die OSI-Netzwerkmanagement-Umgebung, die ein Teilgebiet der OSI-Umgebung *(OSI environment, OSIE)* ist, beinhaltet alle Werkzeuge und Dienste, die für die Kontrolle und Überwachung von Verbindungsaktivitäten benötigt werden. Die OSI-Management-Umgebung erlaubt Managern, Daten zu sammeln, Kontrollen auszuüben, das Vorhandensein der Ressourcen in der OSI-Umgebung (OSIE) und deren Status zu prüfen [ISO7498-4].

Die einzelnen offenen Systeme innerhalb der OSI-Umgebung (OSIE) können Aspekte der Management-Verantwortung realisieren. Die Verantwortung kann durch ein autonomes Management eines offenen Systems oder als Zusammenarbeit mit anderen offenen Systemen durch Austausch von Informationen als koordinierte Managementaktivität realisiert werden. Diese Management-Verantwortung ist auf einzelne Ressourcen gerichtet, die bis zu einem gewissen Grad unabhängig von anderen Ressourcen operieren können. Sie kann zur Koordinierung und Steuerung um eine Reihe von Ressourcen erweitert werden. Diese Erweiterung ermöglicht eine Erhöhung der Funktionalität und Leistung.

#### 2.4.2.3 Verwaltetes Objekt

Ein verwaltetes Objekt stellt die Sicht des OSI-Netzwerkmanagements von einer vom Management überwachten und gesteuerten Ressource in der OSI-Umgebung (OSIE) dar. Das verwaltete Objekt wurde geschaffen, um die Vielzahl unterschiedlicher Ressourcen in einem offenen System einheitlich beschreiben zu können. Ressourcen, die nicht unbedingt OSI-Ressourcen sein müssen, können aus dem Hardwarebereich, zum Beispiel Protokoll-Maschinen[2], Verbindungen, Modems, Workstations, Multiplexer usw., oder aus dem Softwarebereich, zum Beispiel Warteschlange-Programme, Routing-Algorithmen, Puffer-Management-Routinen usw., stammen.
Ein verwaltetes Objekt ist die abstrakte Sicht von einer Ressource, die nur die Eigenschaften repräsentiert, die zum Zweck des Managements notwendig sind. Diese verwalteten Objekte haben bestimmte Eigenschaften, die sie voneinander unterscheiden läßt. Diese Eigenschaften werden Attribute genannt. Die Attribute dienen der Beschreibung der Betriebsmerkmale, des gegenwärtigen Zustands und der Operationsbedingungen eines Objekts [Blac91].

Ein wesentlicher Teil der Definition eines verwalteten Objekts ist die Beziehung zwischen den Eigenschaften der vom verwalteten Objekt repräsentierten Ressource und dem Betriebsverhalten der Ressource. Ein verwaltetes Objekt wird durch die Attribute, die es besitzt, die auf ihm ausführbaren Operationen, die von ihm aussendbaren Meldungen und seine Beziehung zu den anderen verwalteten Objekten festgelegt. Die Attribute besitzen einen Wert, der aus einer einfachen oder komplexen Struktur bestehen kann. Die Ausführung einer Management-Operation darf vom Zustand des verwalteten Objekts

---

2 Siehe „2.3 OSI-Netzwerkmanagement-Terminologie" auf Seite 12

oder ihrer Attribute abhängen. Ein wichtiger Bestandteil der Definition einer Management-Operation ist die Aufzählung der möglichen Formen des Operationsausfalls. Während die Kommunikationsmechanismen der Management-Operation und -Meldungen Bestandteil des Normungsprozesses sind, unterliegen ihre Ausführungsmechanismen keiner Normung.

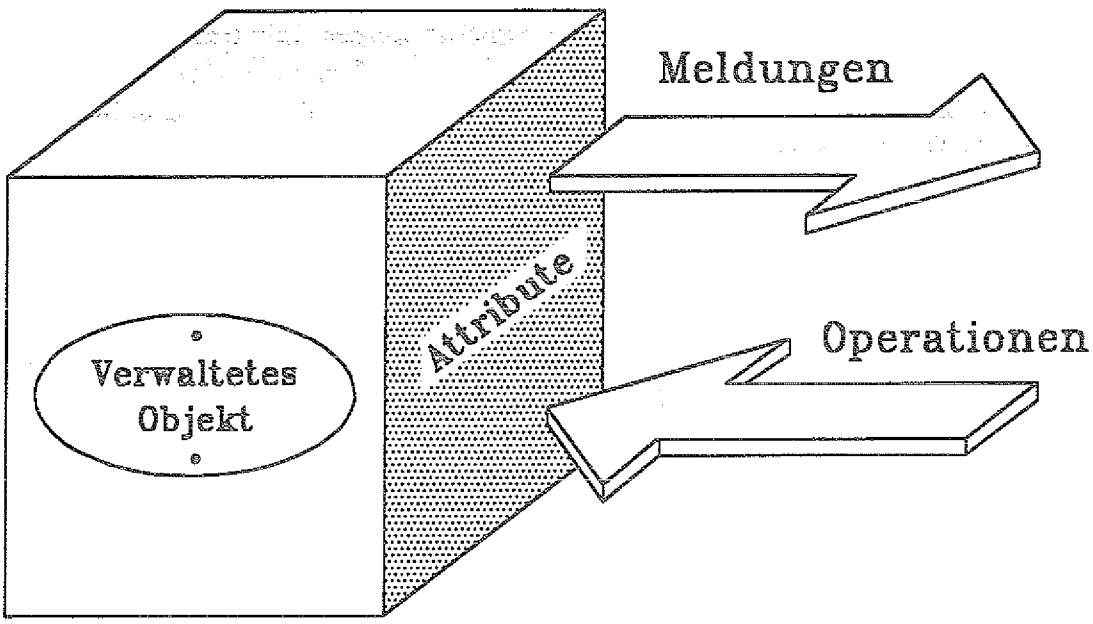



**Abb. 7. Verwaltetes Objekt:** Ein verwaltetes Objekt ist die abstrakte Sicht von einer Ressource, die nur die zum Management erforderlichen Eigenschaften repräsentiert. Die Definition eines verwalteten Objekts erfolgt durch die Festlegung der Attribute, Meldungen, Operationen und der Beziehung zu den anderen verwalteten Objekten.

Eine Management-Operation oder -Meldung kann bestätigt *(confirmed)* oder unbestätigt *(unconfirmed)* sein. In einem realen System existieren verschiedene Instanzen gleichen Typs, die zur einfacheren Kennzeichnung zu einer Klasse (*managed object class*) zusammengefaßt werden. Unter einer Klasse von verwalteten Objekte versteht man eine Menge von verwalteten Objekten, die sich in einem Teil der Attribute, Meldungen und Management-Operationen gleichen (Abb. 7). Zwischen Objekten und Attributen können die beiden folgenden Beziehungen existieren, die innerhalb des Informationsmodells definiert worden sind:

- Vererbungs-Beziehung *(inheritance relationship)* und
- Enthaltungs-Beziehung *(containment relationship)*.

Sonstige Beziehungen zwischen Objekten und Attributen werden im allgemeinen vom OSI-Netzwerkmanagement wahrgenommen, aber sie sind nicht Bestandteil des Informationsmodells [ISO10040].

**Vererbungs-Beziehung** Die Vererbungs-Beziehung bedeutet, daß eine Klasse die Eigenschaften einer übergeordneten Klasse übernimmt (erbt). Eine neue Klasse entsteht, wenn weitere Eigenschaften hinzugefügt werden. In diesem Zusammenhang werden die

Ausdrücke *superclass* und *subclass* verwendet, die jeweils die übergeordnete und die abgeleitete Klasse kennzeichnen.

**Enthaltungs-Beziehung** Eine Beziehung, bei der ein verwaltetes Objekt einer Klasse die verwalteten Objekte der gleichen oder einer anderen Klasse enthalten kann, wird Enthaltungs-Beziehung genannt. Die Ausdrücke *subordinate* und *superior* werden jeweils zur Kennzeichnung der verwalteten Objekte benutzt, die in den anderen enthalten sind und die die anderen enthalten. Ein verwaltetes Subordinate-Objekt ist in einem und nur einem verwalteten Superior-Objekt enthalten. Verwaltete Superior- und Subordinate-Objekte zusammen können auch in einem anderen verwalteten Objekt enthalten sein. Die Enthaltungs-Beziehung beinhaltet die Möglichkeit der hierarchischen Benennung von Objekten. Ein enthaltenes Objekt erhält seinen Namen über den Namen des unmittelbar übergeordneten Objekts. Die Bezeichnung der Objektklasse des enthaltenen Objekts und der Name der speziellen Instanz, welche das Objekt innerhalb dieser Klasse darstellt, vervollständigen die hierarchische Benennung dieses Objekts.

Ein Beispiel, das nur den hierarchischen Aufbau aufzeigt, wird hier zur Benennung eines Modems innerhalb eines Netzwerks erläutert. Es wird angenommen, daß das Netzwerk der AT&T-Klasse angehört und in dieser Klasse mit der Zahl 2 spezifiziert ist. Der Name dieses Netzwerks lautet also $\{AT\&T, 2\}$. Dieses Netzwerk besitzt unter anderen einen Knoten, der der Klasse DCS angehört, und die Adresse 5 hat. Dieser Knoten besitzt jetzt den relativen Namen $\{DCS, 5\}$ und den absoluten Namen $\{AT\&T, 2\}\{DCS, 5\}$. Um jetzt das dritte Modem der Modemklasse 5 mit dem relativen Namen $\{Modem5, 3\}$, das an diesem Knoten angeschlossen ist, zu benennen, wird der absolute Name

$$\{AT\&T, 2\}\{DCS, 5\}\{Modem5, 3\}$$

verwendet. Ein anderes Modem $\{Modem8, 2\}$ kann über einen Multiplexer mit dem relativen Namen $\{Multiplexer, 7\}$ an diesem Knoten angeschlossen sein. Dieses Modem bekommt dann den absoluten Namen:

$$\{AT\&T, 2\}\{DCS, 5\}\{Multiplexer, 7\}\{Modem8, 2\}.$$

Die Enthaltungs-Beziehung definiert das statische und dynamische Verhalten von verwalteten Superior- und Subordinate-Objekten. Ein Beispiel für das statisches Verhalten ist die Anzahl oder die Klasse der verwalteten Objekte, die in einem verwalteten Objekt enthalten sind. Als Beispiel für dynamisches Verhalten sind die Beschränkungen zwischen Attributwerten in verwalteten Superior- und Subordinate-Objekten oder die Verfügbarkeit der enthaltenen und enthaltenden verwalteten Objekte zu nennen [ISO10165-1].

#### 2.4.2.4 Beziehungen zwischen den offenen Systemen

Das OSI-Netzwerkmanagement kann einerseits lokale Operationen ausüben, andererseits kann es Informationen zwischen offenen Systemen austauschen oder auch beide Verfahren mischen, um den Anforderungen der Benutzer gerecht zu werden. Eine Managementaufgabe wird durch die Kooperation der beteiligten Komponenten ausgeführt. Dabei nimmt zum Beispiel eine Komponente die Überwachungsrolle *(managing role)* an, und die anderen die Rolle der überwachten *(managed role)*. Die Rollen können

statisch sein oder sich mit der Zeit ändern. Sie können sich auch nach der speziellen Management-Kommunikation richten. Der Informationsfluß zwischen den offenen Systemen ist im OSI-Netzwerkmanagement durch die Ausübung von Operationen und den Empfang von Meldungen festgelegt.

## 2.5 OSI-Netzwerkmanagement-Modell

Das OSI-Netzwerkmanagement-Modell ist im Management-Rahmenwerk [ISO7498-4] definiert worden. Das Modell befaßt sich mit

- der Struktur des OSI-Netzwerkmanagements,
- der durch OSI-Netzwerkmanagement benötigten unterstützenden Funktionalität,
- dem Fluß von Steuerungsinformationen zwischen den Prozessen und
- dem Fluß von Informationen zwischen den Instanzen.

Das OSI-Netzwerkmanagement umfaßt die Aktivitäten, die zur Steuerung, Koordination und Überwachung der Ressourcen notwendig sind. Unter Ressourcen sollen die Ressourcen verstanden werden, die es erlauben, daß eine Kommunikation in der OSI-Umgebung (OSIE) stattfindet. Die Aktivitäten versetzen ein offenes System in die Lage, eigene Kommunikations-Ressourcen zu überwachen und zu steuern oder in Kooperation mit anderen offenen Systemen die Überwachung und Steuerung der OSI-Umgebung zu übernehmen. Die Ressourcen werden aus der Management-Sicht unter der Bezeichnung verwaltetes Objekt einheitlich dargestellt. Die Begriffe verwaltetes Objekt und Management-Informationsbasis (MIB) beschreiben die in einem offenen System zu verwaltenden Objekte.

Das Management wird durch die Management-Prozesse realisiert. Die Management-Prozesse brauchen nicht in einem bestimmten Ort lokalisiert zu sein und können über viele Systeme verteilt werden. Sind sie nicht in einem System lokalisiert, so müssen sie miteinander kommunizieren, um die Transparenz bezüglich der räumlichen Verteilung zu ermöglichen. Die OSI-Netzwerkmanagement-Protokolle regeln die Kommunikation zwischen den Prozessen in verschiedenen Systemen.

Das OSI-Netzwerkmanagement wird durch drei Formen des Informationsaustausches realisiert:

1. System-Management *(systems management)*;

2. (N)-Schicht Management *((N)-layer management)*;

3. (N)-Schicht Operation *((N)-layer operation)*.

Sie stellen Management-Mechanismen zur Verfügung, die zur Erfüllung der Management-Aufgaben verwendet werden. Die Mechanismen folgen einem hierarchischen Aufbau und besitzen unterschiedliche Wirkungs- und Gültigkeitsbereiche (Abb. 8). Der Management-Informations-Austausch wird durch die Anwendungsschicht oder die (N)-Schicht-Dienste realisiert. Diese Dienste können normale (N)-Dienste oder speziell für Management-Zwecke gelieferte Dienste sein. Abhängig von dem Initiator und den bereitgestellten Diensten kann der Informations-Austausch zwi-

schen zwei oder N Beteiligten stattfinden. Jeder Beteiligte kann die Rolle des Initiators übernehmen, während die anderen dann in die Rolle des Beantworters versetzt werden. Ein Austausch ist z.B. dann erforderlich, wenn Management-Operationen oder -Meldungen ausgeführt werden sollen.

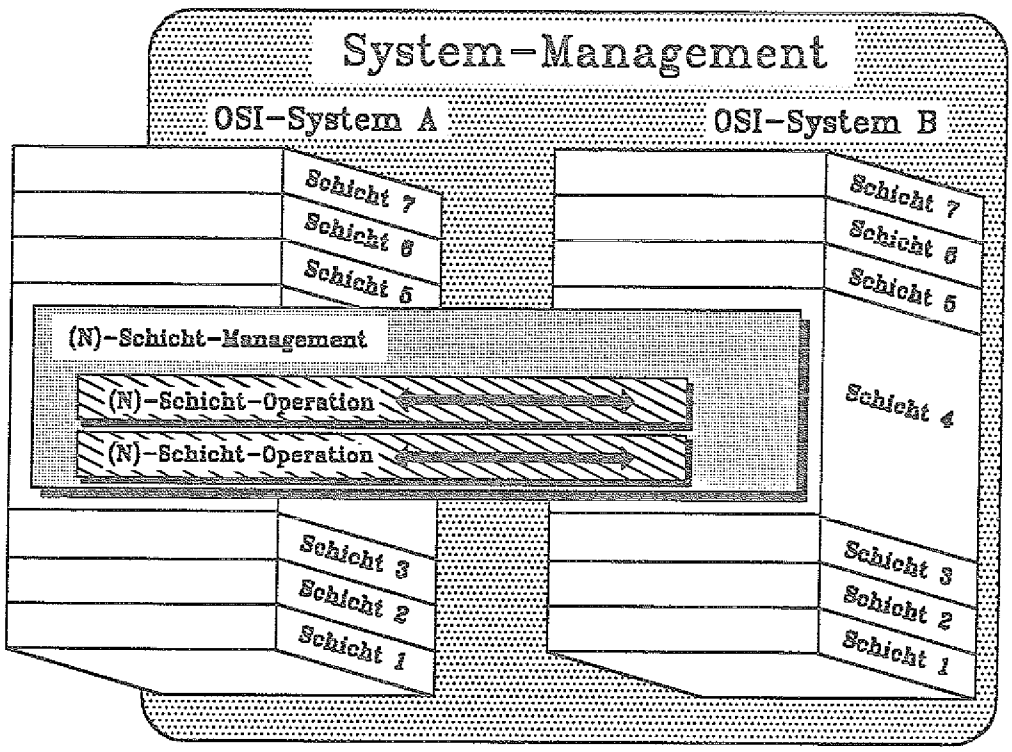



**Abb. 8. Die Kommunikation im OSI-Netzwerkmanagement:** Drei Kommunikationsformen sind für das OSI-Netzwerkmanagement definiert:

1. **System-Management** ermöglicht die Kommunikation mit allen Instanzen des Systems;
2. **(N)-Schicht-Management** ist in der Kommunikation auf die (N)-Schicht begrenzt;
3. **(N)-Schicht-Operation** ist für die Kommunikation mit einer einzelnen Instanz vorgesehen.

### 2.5.1 System-Management

Das System-Management *(systems management)* stellt Mechanismen zur Verfügung, die unter Zuhilfenahme von System-Management-Protokollen der Anwendungsschicht der Überwachung, Steuerung und Koordination von verwalteten Objekten dienen. Das System-Management darf zur Überwachung jedes Objekts innerhalb oder in Verbindung mit einem offenen System benutzt werden. Die vom System-Management zur Verfügung gestellten Mechanismen unterstützen die Funktionalität des gesamten OSI-Netzes. Alle Aufgaben, die mehrere Schichten betreffen oder keiner bestimmten Schicht zugeordnet werden können, unterliegen dem System-Management.

Beispiele für die Aufgaben, die sich auf mehrere Schichten beziehen, sind alle Veränderungen oder das Lesen von Parametern mehrerer Schichten und die Veränderung der System- oder Netz-Konfiguration. System-Management besitzt also einen weiten Einsatzbereich, der die verteilten Verarbeitungs- und Kommunikationsumgebungen umfaßt. Diese Umgebungen spannen einen Bereich von lokalen Netzwerken, die kleine Systeme miteinander verbinden, bis zu den international verbundenen, globalen Netzwerken auf [ISO10040].

Die System-Management-Standards werden in drei Hauptgruppen unterteilt:

1. Standards, die System-Management-Funktionen spezifizieren;

2. Standards, die mit der Spezifikation von verwalteten Objekten in Beziehung stehen;

3. Standards, die die Anwendungs-Schicht-Dienste und -Protokolle für die Übertragung von Informationen in Verbindung mit den Management-Funktionen normieren.

Die Forderungen, die durch System-Management-Aktivitäten erfüllt werden sollen, werden in fünf Bereiche unterteilt:

1. Fehler-Management *(fault management)*,
2. Konfigurations-Management *(configuration management)*,
3. Abrechnungs-Management *(accounting management)*,
4. Leistungs-Management *(performance management)* und
5. Sicherheits-Management *(security management)*.

Diese Bereiche werden im OSI-Netzwerkmanagement-Rahmenwerk [ISO7498-4] definiert. Jeder dieser Bereiche gibt Anlaß zu einem oder mehreren Standards, die eine oder mehrere Funktionen umfassen. Viele Teile der Information, wie z.B. die mit ihnen verbundenen Management-Operationen und die Kommunikations-Protokolle, können mehrere Bereiche gemeinsam benutzen. Zur Ausführung einer Management-Aktivität können mehrere Management-Funktionen kombiniert werden, um eine besondere Management-Strategie zu bewirken [ISO10040].

#### 2.5.1.1 System-Management-Modell

Das System-Management-Modell hat die Aufgabe, die verschiedenen Konzepte des System-Managements zu beschreiben und die Beziehungen zwischen ihnen zu erklären. Das System-Management-Modell stellt die folgenden Aspekte dar:

- Informations-Aspekte,
- Funktions-Aspekte,
- Kommunikations-Aspekte und
- Organisations-Aspekte.

Aus der Sicht der OSI wird das Management einer Kommunikations-Umgebung als eine Anwendung *(application)* betrachtet. Da diese vom Management überwachte Umgebung verteilt ist, sind die Komponenten der Management-Aktivitäten selbst verteilt. Durch die Zusammenwirkung mehrerer Management-Anwendungsinstanzen wird eine

verteilte Management-Anwendung realisiert. Die Wechselwirkung zwischen System-Management-Anwendungsinstanzen wird abstrakt durch Operationen und Meldungen dargestellt. Um die zwischen ihnen auszutauschenden Operationen und Meldungen zu realisieren, werden Dienste und Protokolle benötigt (Abb. 9).

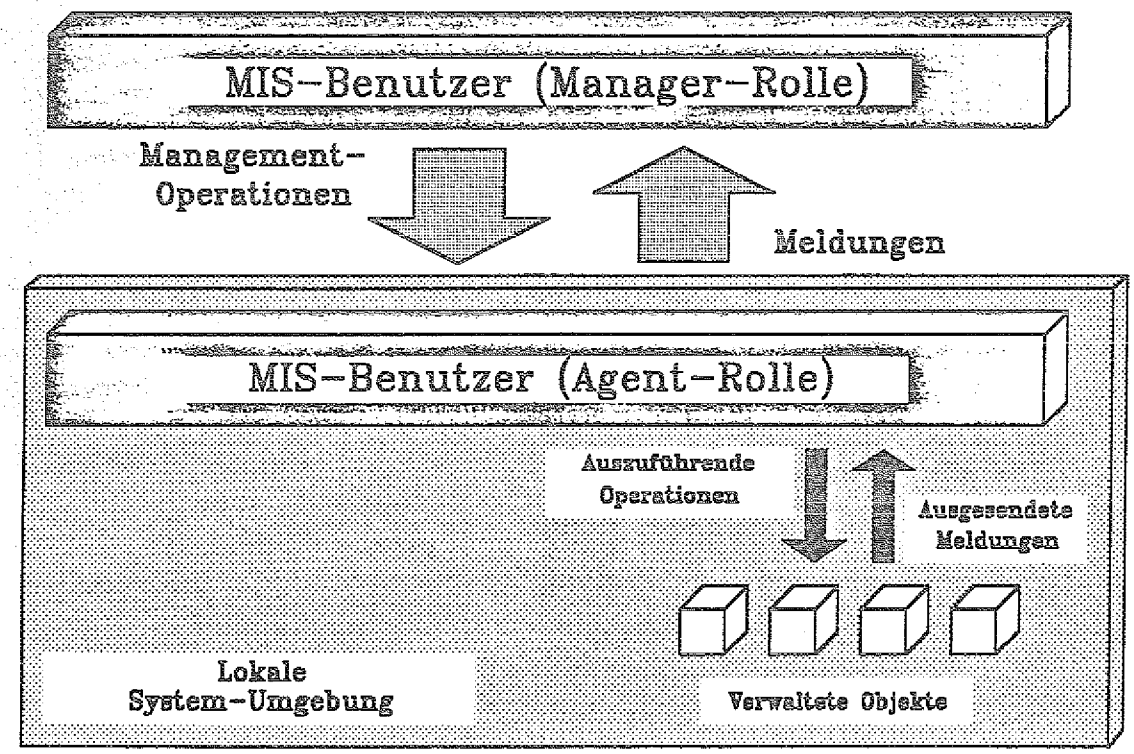



**Abb. 9. Wechselwirkungen innerhalb System-Management:** Die verwalteten Objekte in der lokalen System-Umgebung werden durch den MIS-Benutzer *(Management Information System User)* in der Agent-Rolle verwaltet. Auf die verwalteten Objekte werden Management-Operationen ausgeübt, auf die die verwalteten Objekte in Form von Meldungen antworten. Der MIS-Benutzer in der Agent-Rolle operiert im Auftrag eines MIS-Benutzers in der Manager-Rolle mittels Management-Operationen und leitet die Auftragsresultate, die er von den verwalteten Objekten in Form von Meldungen empfangen hat, an den Auftraggeber weiter.

Die Management-Aktivitäten werden durch die Handhabung von verwalteten Objekten realisiert. In einem offenen System existieren verschiedene Anwendungen. Aus der System-Management-Sicht werden Management-Anwendungen als MIS-Benutzer kategorisiert. Der MIS-Benutzer ist eine Anwendung, die die System-Management-Dienste benutzt. Ein MIS-Benutzer kann die

- Manager-Rolle *(manager role)* oder die
- Agent-Rolle *(agent role)*

übernehmen. In der Manager-Rolle ist er fähig, Management-Operationen auszuüben und Meldungen zu empfangen. Unter einem Manager versteht man in OSI ein MIS-Benutzer, der für einen speziellen Austausch von System-Management-Informationen die Manager-Rolle übernommen hat. Er ist der Teil einer verteilten An-

wendung, der für eine oder mehrere Management-Aktivitäten die Verantwortung trägt (Abb. 10).

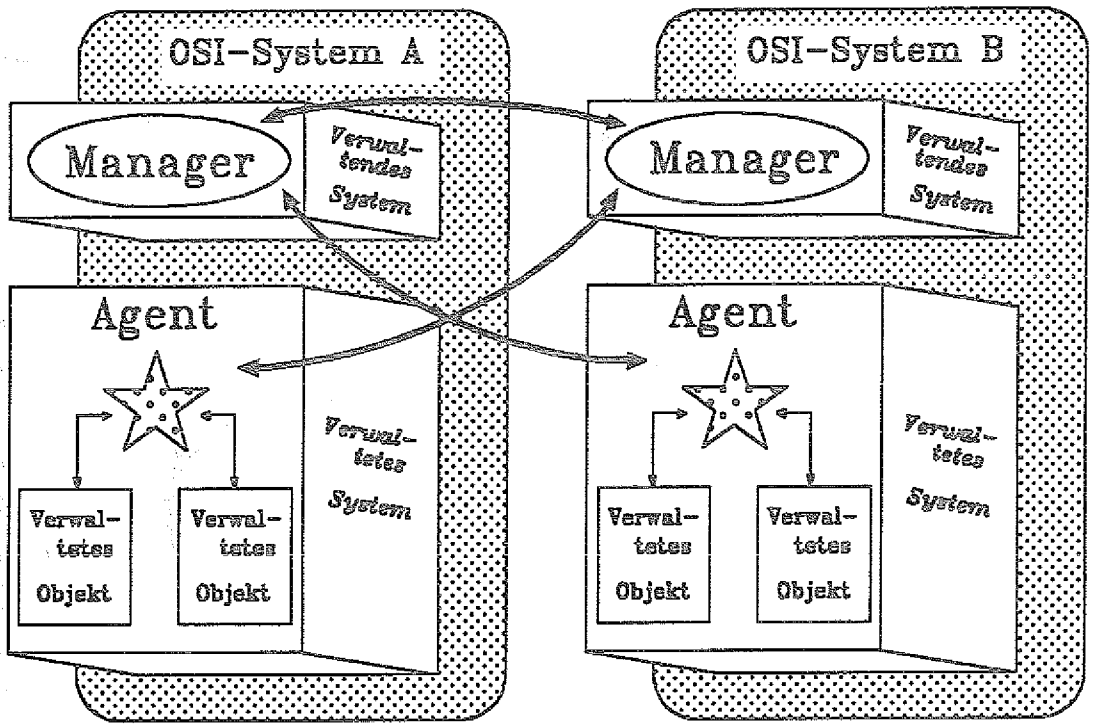



**Abb. 10. Die Rollen innerhalb eines OSI-Systems:** Ein OSI-System kann einen MIS-Benutzer in zwei Formen unterstützen. Wird ein MIS-Benutzer in der Manager-Rolle (Manager) unterstützt, heißt das OSI-System ein verwaltendes System. Wird ein MIS-Benutzer in der Agent-Rolle unterstützt, wird das OSI-System ein verwaltetes System genannt. Die Abbildung verdeutlicht die Rollenverteilung und dadurch das dezentrale Netzwerkmanagement.

Ein MIS-Benutzer in der Agent-Rolle ist in der Lage, Operationen auf den verwalteten Objekten auszuführen und Meldungen über verwaltete Objekte auszusenden. Die Anweisungen hierzu werden ihm durch einen MIS-Benutzer in der Manager-Rolle erteilt. Er leitet die ausgesendeten Meldungen von verwalteten Objekten dem MIS-Benutzer in der Manager-Rolle weiter. Ein Agent ist nach ISO ein MIS-Benutzer, der die Agent-Rolle übernommen hat, um spezielle System-Management-Informationen auszutauschen. Er ist der Teil einer verteilten Anwendung, der die verwalteten Objekte innerhalb ihrer lokalen Systemumgebung überwacht. Das Manager-Konzept ist nicht nur auf die Anwendungen beschränkt, die am System-Management beteiligt sind, sondern alle Anwendungen, die den Zugriff auf Management-Informationen benötigen, dürfen System-Management-Dienste benutzen [ISO10040].

Ein MIS-Benutzer in der Manager-Rolle spezifiziert, ob eine Operation eine Bestätigung braucht, während ein MIS-Benutzer in der Agent-Rolle bestimmt, ob eine Meldung eine Bestätigung benötigt. Ein verwaltetes System-Objekt ist ein verwaltetes Objekt, das für mehrere Schichten, für eine spezifische System-Management-Funktion oder für das ganze System relevant ist.

Das *system management overview* [ISO10040] ist ein konzeptionelles Modell und stellt die Struktur und Komponenten der Information dar, die tatsächlich durch die standardisierten Management-Informationsdienste ausgetauscht werden. Wenn vom Austausch der Management-Information gesprochen wird, ist dabei dieses Modell gemeint.

#### 2.5.1.2 Informations-Aspekte

Die Informations-Aspekte des System-Management-Modells, beschrieben im [ISO10040], befassen sich mit der Verfeinerung des Konzeptes der verwalteten Objekte[3], das im Management-Rahmenwerk [ISO7498-4] erläutert worden ist, mit ihren Attributen, mit den auf ihnen ausgeübten Operationen und mit den von ihnen ausgesendeten Meldungen. Eine vollständige Beschreibung des Informationsmodells wird im *Management Information Model* [ISO10165-1] dargestellt. Die Richtlinien für die Definition von verwalteten Objekten sind im *Guidelines for the Definition of Managed Objects* [ISO10165-4] festgelegt worden. Eine Sammlung von Definitionen der Management-Information, die bei der Verwaltung erlaubt sind, sind im *Definition of Management Information* [ISO10165-2] aufgeführt.

Die im OSI-Netzwerkmanagement ausgetauschten Informationen dienen der Überwachung von verwalteten Objekten. Sie werden in die drei folgenden Typen unterteilt:

- Daten *(data)*: Dieser Typ stellt die Informationen dar, die ein Initiator von einem Beantworter empfängt. Als Beispiel kann die Information über den Zustand einer Verbindung genannt werden, ob sie sich also im Betrieb oder im Wartezustand befindet.

- Steuerung *(control)*: Zwei Benutzer eines OSI-Netzwerkmanagement-Dienstes benutzen diesen Typ, um die Attribute oder den Zustand eines verwalteten Objekts zu ändern. Die Steuerungsinformationen, die den Übergang vom Wartezustand zum Betriebszustand ermöglichen, sind ein Beispiel für diesen Typ.

- Ereignisse *(events)*: Ein Benutzer wird von einem eingetretenen Ereignis im System durch diesen Typ von Informationen benachrichtigt. Normalerweise wird die Zeit des Auftretens dieses Ereignisses berichtet. Dem Benutzer wird beispielsweise zusammen mit der Zeit des Geschehens gemeldet, daß eine Zustandsänderung bei der Verbindung stattgefunden hat.

Die Informationen erreichen das System-Management aus

1. der lokalen Eingabe,
2. einem anderen offenen System durch Kommunikation über die Anwendungsschicht oder
3. den unteren Schichten.

---

[3] siehe auch „2.4.2.3 Verwaltetes Objekt“ auf Seite 17

### 2.5.1.3 Funktions-Aspekte

Die Funktions-Aspekte des System-Management-Modells, beschrieben im System-Management-Überblick *systems management overview* [ISO10040], führen die Konzepte der spezifischen Management-Funktionsbereiche *(specific management functional areas, SMFAs)* ein, die früher als *specific management information services and protocols (SMIS/SMIP)* bezeichnet wurden [Kauf90a].

Die OSI benutzt die Forderungen der Anwender, um einen Referenzrahmen zu schaffen, der die Einordnung der Forderungen in fünf Bereiche ermöglicht:

1. *Fehler-Management*: umfaßt die Aufgabenbereiche Fehlererkennung, Fehlerisolierung und Fehlerbehebung, wobei in einem offenen System gemäß OSI-Netzwerkmanagement Fehler durch die von verwalteten Objekten ausgesendeten Fehler-Meldungen sichtbar werden [Rose90b]. Die für die obigen Aufgaben benutzten Funktionen sollen im wesentlichen die folgenden Dienste erbringen:

   - „Wartung und Prüfung von Fehler-Logs,
   - Annahme und Agieren auf Fehler-Meldungen,
   - Ablaufverfolgung und Identifikation von Fehlern,
   - Ausführen einer Folge von Diagnose-Tests und
   - Korrektur von Fehlern“[4].

2. *Sicherheits-Management*: umfaßt die Forderungen, die dem Schutz der zur Kommunikation benötigten Ressourcen dienen. Dazu werden Funktionen benötigt, die im wesentlichen die folgenden Dienste zur Verfügung stellen:

   - „Erzeugung, Löschung und Steuerung von Sicherheits-Diensten und Mechanismen,
   - Verteilung von sicherheitsrelevanten Informationen und
   - Meldung von sicherheitsrelevanten Ereignissen“ [ISO7498-4].

3. *Leistungs-Management*: ermöglicht die Überwachung des Verhaltens von OSIE-Ressourcen und der Wirksamkeit der Kommunikationsaktivitäten, d.h. die Leistung des Netzes wird beobachtet, und im Falle eines Leistungsabfalls werden korrektive Maßnahmen durch Funktionen, die durch nachfolgende Aktivitäten angedeutet werden, eingeleitet:

   - „Sammeln von statistischen Informationen,
   - Wartung und Prüfung von Logs, in denen die vergangenen Zustände des Systems registriert sind,

---

[4] siehe Seite 3 in [ISO7498-4]

- Bestimmung der System-Leistung unter natürlichen und künstlichen Zuständen und

- Änderung der Systembetriebsarten zum Zweck der Einleitung von Leistungs-Management-Aktivitäten" [ISO7498-4].

4. ***Abrechnungs-Management***: unterstützt die Spezifizierung der durch die Benutzung von Ressourcen entstandenen Kosten und die zu entrichtenden Gebühren für die Benutzer. Die dazu benötigten Funktionen sollen folgendes leisten:

   - „Unterrichtung der Benutzer von entstandenen Kosten oder verbrauchter Ressourcenleistung,

   - Fähigkeit, Abrechnungsgrenzen festzulegen und Tarife bezüglich verbrauchter Leistung zu bestimmen und

   - Fähigkeit der Kombination von Kosten, wenn mehrere Ressourcen zur Erfüllung einer gegebenen Aufgabe benutzt werden" [ISO7498-4].

5. ***Konfigurations-Management***: dient der Verwaltung im Sinne von Identifikation und Steuerung von offenen Systemen sowie Datenaustausch zwischen offenen Systemen, um Initialisierung, Inbetriebnahme, kontinuierlichen Betrieb und Beendigung von Kommunikationsdiensten zu ermöglichen. Dazu werden Funktionen mit folgenden Diensten benutzt:

   - „Setzen von Parametern, die den Routinebetrieb eines offenen Systems steuern,

   - Zuordnung von Namen zu verwalteten Objekten und Gruppen von verwalteten Objekten,

   - Initialisierung und Stillegung von verwalteten Objekten,

   - Auf Anfrage Sammeln von Information über den augenblicklichen Zustand des offenen Systems,

   - Erhaltung von Informationen über signifikante Änderungen im Zustand des offenen Systems und

   - Änderung der Konfiguration des offenen Systems" [ISO7498-4].

Innerhalb dieses Rahmens werden die Funktionen definiert. Der Zweck dieser Vorgehensweise ist eine Annäherung an eine anwenderorientierte Definition der Funktionen. Die zur Erfüllung der Management-Forderungen notwendigen Management-Aktivitäten und -Informationen werden durch die Spezifikation der System-Management-Funktionen definiert. Die Management-Funktionen dürfen miteinander kombiniert werden, um eine spezifische Management-Aktivität durchzuführen. Jede Forderung wird durch eine oder einige Funktionen erfüllt. Und umgekehrt darf eine Funktion durch mehrere Forderungen aufgerufen werden. Dadurch entsteht eine viele-zu-viele Beziehung *(many-to-many relationship)* zwischen den Funktionen und den existierenden Forderungen [ISO10040].

Die Forderungen, aus der Perspektive der Funktionsbereiche gesehen, führen zu den Standards für die Definition der allgemeinen Management-Aktivitäten und -Informationen. Diese Standards beinhalten:

- Funktionen, die zur Erfüllung der Aufgaben der Funktionsbereiche erforderlich sind,
- Mechanismen, die mit der Bereitstellung der Funktionen in Verbindung stehen,
- Benutzung der unterstützenden OSI-Dienste zur Bereitstellung der notwendigen Kommunikation,
- Klassen von verwalteten Objekten, auf denen die Mechanismen operieren, um die gewünschte Funktion bereitzustellen und
- Anforderungen nach der Konformität zu den Standards.

Bei einer Verbindung werden nicht immer alle Dienste angefordert, die eine System-Management-Funktion zur Verfügung stellt. Aus diesem Grund werden die Dienste der System-Management-Funktion in Funktionseinheiten *(functional units)* unterteilt. Sie gelten als die Bausteine der Verhandlung *(negotiation)* zwischen den MIS-Benutzern. Es dürfen Funktionseinheiten definiert werden, die eine größere Anzahl von Diensten im Vergleich zu einer Funktion liefern. Diese Überschreitung der Funktionsgrenzen soll stattfinden, um Meldungen, Management-Operationen oder beide zusammen zu unterstützen. Die anderen Funktionseinheiten werden zum Aushandeln von Teilmengen dieser Fähigkeiten, wie Überwachung und Steuerung, definiert. Ein Agent kann im allgemeinen nicht bestimmen, wofür die Management-Operationen, die er empfängt, oder die Meldung, die er aussendet, vorgesehen sind. Er antwortet auf die Anforderung des Managers individuell, ohne einen umfangreichen Kontext zur Ausführung der Anforderung des Managers zu benötigen. Zum Beispiel kann in einem offenen System nicht unterschieden werden, ob die Aufforderung zum Lesen des Fehler-Zählers dem Fehler-Management oder dem Leistungs-Management dient.

#### 2.5.1.4 Kommunikations-Aspekte

Das Manager-Agent-Konzept ist das von den MIS-Benutzern für die Kommunikation verwendete Konzept. Die Kommunikation wird durch die OSI-Protokolle geregelt. CMIS[5] *(Common Management Information Service)* ist der allgemeine Kommunikations-Dienst für das System-Management und unterstützt das Manager-Agent-Konzept. Der CMIS ist im *common management information service definition* [ISO9595] normiert worden und ist ein Anwendungs-Dienst auf der Anwendungsschicht. Die MIS-Benutzer dürfen auch andere Anwendungs-Dienste benutzen, wie *file transfer, access and management (FTAM)*. Es ist nicht erforderlich, daß die genannten Anwendungen auch das Manager-Agent-Konzept trotz der Unterstützung der MIS-Benutzer unterstützen.
Die durch die MIS-Benutzer zur Kommunikation verwendeten Management-Operationen und Meldungen werden unterstützt. Diese Unterstützung bezieht sich einerseits auf die Übertragung der Anforderungen zwischen den MIS-Benutzern, an-

---

5 siehe „2.5.5.1 Allgemeiner Management-Informationsdienst" auf Seite 39

dererseits auf die Steuerung des Zugriffs auf die verwalteten Objekte und die Verbreitung von Meldungen nach außen.

Die verschiedenen Management-Operationen sind im *structure of management information - Part 1: Management information model* [ISO10165-1] beschrieben. Die System-Management-Dienste besitzen sowohl Dienstelemente[6], die die Kommunikation in Verbindung mit den Anforderungen für verschiedene Management-Operationen ermöglichen, als auch Dienstelemente, die die Meldungsinformationen übertragen. Hierdurch wird der an der Grenze von verwalteten Objekten definierte Austausch spezifiziert. Zusätzliche Unterstützung für die Identifikation der verwalteten Objekte mittels Filterung und für die Steuerung der Kommunikation werden durch die System-Management-Dienste geliefert.

Eine in das verwaltete System eintreffende Anforderung für Management-Operationen kann nur unter Berücksichtigung des Zugriff-Steuerungs-Mechanismus *(access control mechanismus)* angenommen werden. Die Meldungen der verwalteten Objekte werden durch einen Mechanismus nach bestimmten Kriterien geprüft und danach zur Kommunikation nach außen frei gegeben. Die Struktur des System-Managements wird definiert durch

- den System-Management-Anwendungsprozeß (SMAP) und
- die System-Management-Anwendungsinstanz (SMAE).

Der System-Management-Anwendungsprozeß *(systems management application process, SMAP)* kennzeichnet die speziellen Anwendungsprozesse, die die System-Management-Mechanismen oder -Funktionen realisieren. Sie besitzen systemübergreifende Informationen von Systemparametern und können daher alle Aspekte des OSI-Systems berücksichtigen, die ein einzelnes System oder verschiedene Systeme betreffen [Robo90]. Die das OSI-Netzwerkmanagement unterstützenden Management-Prozesse erhalten ihre Steuerungsinformationen aus zwei Quellen. Die erste Quelle sind die Elemente, die als verwaltende Agenten lokal den Management-Prozeß steuern. Die zweite Quelle sind die entfernten Systeme, die dem Management-Prozeß die Steuerungsinformationen durch SMAEs, (N)-Schicht-Management-Instanzen und (N)-Instanzen zur Verfügung stellen. Die Management-Prozesse steuern die verwalteten Objekte direkt im eigenen System und indirekt in anderen Systemen mittels Protokoll-Austausch durch SMAEs, (N)-Schicht-Management-Instanzen und (N)-Instanzen. Die abstrakte Syntax und Semantik des Steuerungsflusses innerhalb der OSI-Umgebung wird durch den OSI-Protokoll-Austausch definiert [ISO10040].

Unterstützt ein reales (offenes) System *(real (open) system)* die Manager- oder Agent-Rolle eines MIS-Benutzers, wird es verwaltendes System bzw. verwaltetes System genannt. Dabei soll unter einem realen offenen System ein System verstanden werden, das sich in bezug auf die Kommunikation mit anderen realen Systemen nach OSI richtet. Ein reales (offenes) System kann gleichzeitig ein verwaltetes oder ein verwaltendes System sein und dadurch die Realisierung des dezentralen Netzwerkmanagements ermöglichen. Man braucht dann nicht alle Dienste in allen Systemen realisieren, d.h. Dienst

---

6 siehe „2.2 ISO/OSI-Referenzmodell" auf Seite 5

X kann im System A implementiert sein und vom System B benutzt werden, und Dienst Y kann entsprechend im System B implementiert sein und vom System A benutzt werden (Abb. 10).
Ein MIS-Benutzer als Manager, auch Managing-Prozeß genannt, ist der Teil des System-Management-Anwendungsprozesses, der die verwalteten Objekte anderer Systeme des offenen Netzes verwaltet. Ein MIS-Benutzer als Agent, auch Agent-Prozeß genannt, verwaltet im Auftrag eines Managing-Prozesses die lokal in seinem System vorhandenen Objekte.

Die OSI-Kommunikation, die die System-Management-Funktionen betrifft, wird durch die System-Management-Anwendungsinstanz *(systems management application entity, SMAE)* realisiert. Jeder Anwendungsprozeß, der entsprechend dem Management-Protokoll-Standard kommuniziert, benutzt eine SMAE.

Eine SMAE benötigt in einem offenen System eine ausreichende Unterstützung durch die sieben Schichten, bevor sie ihre System-Management-Funktionalität den anderen offenen Systemen zur Verfügung stellen kann. Fehlt die unterstützende Funktionalität für die SMAEs, so kann ein offenes System nur die Aktivitäten anbieten, die sich aus den einzelnen, individuellen Funktionalitäten der (N)-Schichten innerhalb des offenen Systems zusammensetzen. Dabei wird vorausgesetzt, daß die (N)-Schicht durch die Schichten 1 bis (N-1) genügend unterstützt wird. Steht weder System-Management noch (N)-Schicht-Management zur Verfügung, so ist die größte vom OSI-Netzwerkmanagement angebotene Funktionalität, ein Satz von einzelnen individuellen Funktionalitäten, die die (N)-Schicht-Operation bereitstellt. Die Existenz einer SMAE ist nicht von der Existenz der (N)-Schicht-Management-Instanzen auf irgendeiner Schicht abhängig [ISO7498-4].

Die System-Management-Protokolle auf der Anwendungsschicht regeln die OSI-Kommunikation zwischen den unterschiedlichen System-Management-Instanzen. Die System-Management-Kommunikation stellt die normale Methode zum Austausch von Informationen dar. Diese Kommunikation findet zwischen den System-Management-Anwendungsinstanzen statt. Da die System-Management-Protokolle auf der Anwendungsschicht implementiert sind, brauchen sie die Unterstützung der Protokolle der darunterliegenden Schichten. Realisiert ein System nicht alle sieben Schichten, müssen solche Systeme zur Integration zum OSI-System mit einer minimaler Untermenge von Diensten *(minimum communication capability)* zum Betrieb eines System-Managements ausgestattet werden [Rose89]. Die Anwendungsdienstelemente (ASE)[7] unterstützen das System-Management. Die System-Management-Anwendungsinstanz (SMAE) setzt sich aus den folgenden Komponenten zusammen (Abb. 11):

- SMASE *(Systems Management Application Service Element)*,
- CMISE *(Common Management Information Service Element)*,
- andere ASEs, wie FTAM und TP,
- ACSE *(Association Control Service Element)*,

---

7 siehe „2.3 OSI-Netzwerkmanagement-Terminologie“ auf Seite 12

- ROSE *(Remote Operations Service Element)* und
- SACF *(Single Association Control Function)*.

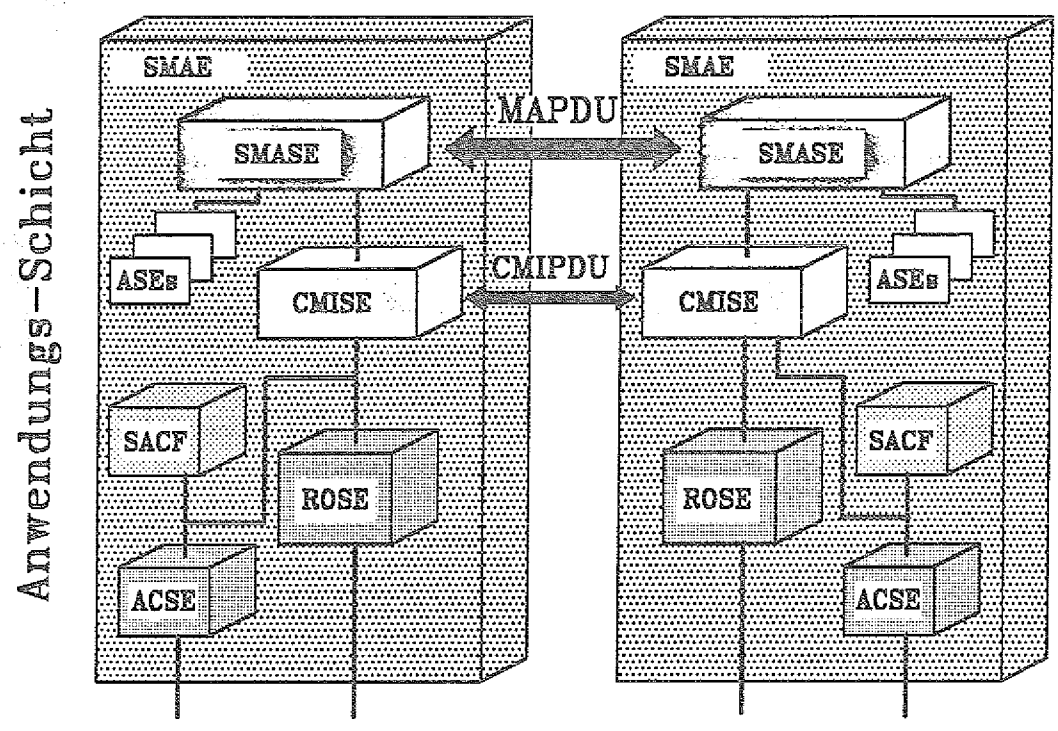



**Abb. 11. OSI-Netzwerkmanagement und die Anwendungsschicht:** Die Abbildung stellt die Komponenten der SMAE dar, die zur Bereitstellung ihrer Funktionalität erforderlich sind. Eine SMAE nimmt die Dienste der darunterliegenden Schichten in Anspruch und kommuniziert mit den SMAEs in anderen Systemen.

Das System-Management-Anwendungsdienstelement (SMASE) definiert die Semantik und die abstrakte Syntax der das OSI-Netzwerkmanagement betreffenden Informationen, die in Management-Anwendung-Protokoll-Dateneinheiten *(Management Application Protocol Data Units, MAPDUs)* übertragen werden. Die vom OSI-Protokoll realisierte MAPDU stellt das abstrakte Konzept der zwischen Management-Anwendungsinstanzen ausgetauschten Management-Operationen und Meldungen dar. Für jede definierte MAPDU wird auch die Anpassung zu den unterstützten Diensten spezifiziert [ISO10040]. Das SMASE beschreibt die spezifische Management-Information, die zwischen den System-Management-Anwendungsinstanzen ausgetauscht werden soll. Das SMASE stellt seinen Benutzern Dienste bereit, indem es selbst die Dienste der anderen ASEs benutzt. Die durch SMASE benutzten Kommunikationsdienste werden durch CMISE, das ein spezielles Anwendungsdienstelement für Management-Zwecke ist, oder andere ASEs, wie FTAM *(File Transfer, Access and Management)* oder TP *(Transaction Processing)*, die jeweils in [ISO8571] und [ISO10026-1] beschrieben sind, bereitgestellt. Das CMISE spezifiziert seinerseits die Dienste und Prozeduren für die Übertragung von CMIP-Dateneinheiten *(Common Management Information Protocol Data Units, CMIPDUs)* und liefert ein Mittel, das den Austausch von Management-

Informationen in Management-Operationen und -Meldungen allgemein implementiert. Die anderen von SMASE benutzten ASEs können auch zum Austausch von Management-Informationen benutzt werden.

Das CMISE nimmt die Dienste von ROSE und ACSE in Anspruch, um eine Kommunikation mit den unteren Schichten zu ermöglichen und ihre Dienste zu benutzen. Das ACSE wird auch von der SACF verwendet, um dem SMAE die Management-Verbindungs-Dienste zur Verfügung zu stellen. Die SMASE, CMISE und ROSE teilen sich eine einzelne abstrakte Syntax, die im *common management information protocol* [ISO9596] definiert ist.

**Anwendungs-Kontext-Verhandlung** Eine Verbindung in der System-Management-Umgebung wird zwischen zwei System-Management-Anwendungs-Instanzen errichtet, indem sie sich auf einen Anwendungs-Kontext *(application context)* einigen. Der Anwendungs-Kontext bestimmt das verteilte Management-Wissen *(shared management knowledge)* für diese Verbindung zusammen mit den zu benutzenden Anwendungsdienstelementen (ASEs). Im System-Management-Überblick *systems management overview* [ISO10040] werden drei Anwendungs-Kontexte definiert:

1. Der Manager-Anwendungs-Kontext *(manager application context)*,
2. der Agent-Anwendungs-Kontext *(agent application context)* und
3. der Manager-Agent-Anwendungs-Kontext *(manager and agent application context)*.

Diese drei Anwendungs-Kontexte stehen für eine Verbindung in einer System-Management-Umgebung zur Verfügung. Die Zuordnung anderer Namen ist möglich, wenn man die Benutzung eines anderen Satzes von ASEs voraussetzt [ISO10040].

Bei jedem der im System-Management definierten Kontexte werden folgende Festlegungen getroffen:

1. Der Satz von ASEs ist bei den drei Anwendungs-Kontexten gleich und setzt sich aus ACSE, ROSE, CMISE, SMASE zusammen.
2. Die Namen der Anwendungs-Kontexte haben die folgenden Objektbezeichner-Werte *(object identifier value)*:
   - Manager-Anwendungs-Kontext[8]

     ```
     {joint-iso-ccitt ms(9) smo(0) application-context(0) manager(0)};
     ```

[8] Die in den geschweiften Klammern stehenden Ausdrücke beschreiben, wie die Objekte innerhalb des OSI-MIB-Baums einzuordnen sind, d.h. wie der MIB-Baum zu durchqueren ist, um das betreffende Objekt zu finden. Die Werte in Klammern stellen die numerischen Werte der Knoten dar, die auf dem Weg zum betreffenden Objekt liegen. Die Syntax erlaubt die nur numerische, nur textliche oder gemischte Bezeichnung von Objekten, wie z.B. hier die gemischte Darstellung verwendet wurde, und deshalb der numerische Wert von joint-iso-ccitt(2) weggelassen worden ist (vgl. Abb. 27 auf Seite 76).

- Agent-Anwendungs-Kontext

  ```
  {joint-iso-ccitt ms(9) smo(0) application-context(0) agent(1)};
  ```

- Manager und Agent-Kontext

  ```
  {joint-iso-ccitt ms(9) smo(0) application-context(0) manager-agent(2)}.
  ```

3. Die Zuordnung der Rollen geschieht entsprechend dem Manager-Agent-Konzept. Bei dem Manager-Anwendungs-Kontext und Agent-Anwendungs-Kontext ist der Initiator der Manager bzw. der Agent und der Beantworter der Agent bzw. der Manager. Der Manager-Agent-Kontext erlaubt, daß der Initiator und der Beantworter beide Rollen übernehmen darf.

4. Die Verbindungs-Informations-Parameter *(association information parameter, [ISO8650])* bestehen für die drei genannten Anwendungs-Kontexte aus einer Folge von External-Daten *(EXTERNAL data, [ISO9596])* für CMISE, gefolgt von External-Daten für SMASE. Die für SMASE gelieferten External-Daten sind vom Daten-Typ *FunctionalUnitPackage* in ASN.1.

Die Regeln zur Errichtung einer Anwendungs-Verbindung durch den Anwendungs-Kontext sind im [ISO10040] erläutert und werden weiter unten zusammengefaßt (vgl. Tab. 1).

- Wenn der Initiator der Verbindung den Manager- bzw. Agent-Anwendungs-Kontext spezifiziert, muß der Beantworter mit demselben Anwendungs-Kontext antworten oder die Errichtung der Verbindung zurückweisen. Im Falle der erfolgreichen Errichtung der Verbindung soll der Initiator die Manager- bzw. Agent-Rolle und der Beantworter die Agent- bzw. Manager-Rolle für alle auf der Verbindung stattfindenden Wechselwirkungen übernehmen.

- Wenn der Initiator den Manager-Agent-Anwendungs-Kontext spezifiziert, muß der Beantworter mit einem der folgenden Kontexte antworten oder die Errichtung der Verbindung ablehnen:

  1. Manager-Agent-Anwendungs-Kontext,
  2. Manager-Anwendungs-Kontext oder
  3. Agent-Anwendungs-Kontext.

  Wenn die Verbindung erfolgreich errichtet wird, und der Beantworter mit Manager-Agent-Anwendungs-Kontext antwortet, können die Rollen nach jeder auf der Verbindung stattfindenden Wechselwirkung ausgetauscht werden. Der Aufrufer der Wechselwirkung entscheidet über die Zuordnung der Rollen. Antwortet der Beantworter nach der erfolgreichen Errichtung der Verbindung mit Manager- bzw. Agent-Anwendungs-Kontext, so soll der Initiator die Rolle des Managers bzw. Agents und der Beantworter die Rolle des Agents bzw. Managers für alle auf der Verbindung stattfindenden Wechselwirkungen übernehmen.

**Funktionseinheit-Verhandlung** Die verhandelnden CMISE-Funktionseinheiten gehorchen den in [ISO9596] definierten Verhandlungsregeln. Die Verhandlung der System-

Management-Funktionseinheiten *(systems management functional units, SMFUs)* ist optional. Wenn aber die SMFUs spezifiziert werden, ist die Verbindung auf die zugelassenen Funktionseinheiten begrenzt. Der Beantworter kann aufgrund der vorgeschlagenen Funktionseinheiten die Verbindung annehmen, ablehnen oder über sie verhandeln.

Wenn der Initiator der Verbindung bei der Anforderung SMFU vorschlägt, hat der Beantworter die Verantwortung, die spezifische Funktionalität zu bestimmen. Dies bedeutet, wenn nur Meldungen durch den Beantworter vorgeschlagen werden, werden die Ereignis-Berichte die einzigen akzeptablen Wechselwirkungen der Verbindung sein.
Werden aber keine SMFUs vorgeschlagen, bedeutet es für die Verbindung, daß keine spezifische Funktionalitäts-Begrenzung vorliegt. Hierdurch besitzt der Beantworter einen großen Spielraum, der sich von einem begrenzten Satz von SMFUs bis zu einem unbegrenzten Satz von Funktionalität erstreckt. Die uneingeschränkte Funktionalität wird erreicht, wenn in der Antwort von A-ASSOCIATE keine SMASEUserData, ein Funktionseinheit-Paket mit Spezifikationen über die Verbindung, zurückgeschickt wird. Es muß aber berücksichtigt werden, daß die Wechselwirkungen auf der Verbindung innerhalb der definierten Grenzen ablaufen sollen, die durch CMIS-Funktionseinheiten und durch den im Anwendungs-Kontext-Namen spezifizierten Anwendungs-Kontext festgelegt sind.

Wenn keine spezifischen SMFUs verhandelt worden sind, d.h. SMASEUserData nicht existieren (UserInfo abwesend), kann jede beliebige Wechselwirkung gewählt werden. Wird die gewünschte Wechselwirkung durch die beiden Management-Systeme nicht unterstützt, führt dies zu einem Fehler. Sind aber spezifische SMFUs verhandelt worden (UserInfo anwesend), führt jede gewünschte Wechselwirkung außerhalb der Grenzen der ausgehandelten SMFUs zu einem Fehler. Die Management-Systeme sind dafür verantwortlich, daß die für jede Verbindung festgelegten Anforderungen und Begrenzungen eingehalten werden.

**Management-Wissen** Das Management-Wissen *(management knowledge)* für die System-Management-Kommunikation umfaßt:

- Das Protokoll-Wissen, wie z.B. den Anwendungs-Kontext,
- das Funktion-Wissen, wie z.B. die Funktionen und Funktionseinheiten,
- das Wissen über verwaltete Objekte, wie z.B. die Klassen, Instanzen und Identifikation von verwalteten Objekten und ihre Attribute und
- die Begrenzungen auf die unterstützten Funktionen und die Beziehung zwischen Funktionen und verwalteten Objekten. Um beispielsweise den spezifischen Funktionen Unterstützung zu liefern, ist die Existenz von für sie wichtigen verwalteten Objekten erforderlich.

Die Existenz des verteilten Management-Wissens ist im Vorhandensein der verteilten Management-Anwendungen begründet. Das verteilte Management-Wissen zwischen zwei offenen Systeme ist von der Rolle der End-Systeme abhängig, da die verbundenen offenen Systeme unterschiedliche verwaltete Objekte beinhalten können und dadurch die jeweilige Sicht des End-Systems bestimmen. Es besteht die Notwendigkeit, verteiltes

Management-Wissen zwischen zwei Systemen, die Management-Informationen austauschen, zu errichten und zu modifizieren.

| Die möglichen Fälle | | | Initiator | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Manager-AK. | | Agent-AK. | | Manager-Agent-AK. | |
| | | | UserInfo anwesend [1)] | UserInfo abwesend | UserInfo anwesend [2)] | UserInfo abwesend | UserInfo anwesend [3)] | UserInfo abwesend |
| Beantworter | Manager-AK. | UserInfo anwesend[4)] | VV | UE | A | A | VV | UE |
| | | UserInfo abwesend | UE | UE | A | A | UE | UE |
| | Agent-AK. | UserInfo anwesend[5)] | A | A | VV | UE | VV | UE |
| | | UserInfo abwesend | A | A | UE | UE | UE | UE |
| | Manager-Agent-AK. | UserInfo anwesend[6)] | A | A | A | A | VV | UE |
| | | UserInfo abwesend | A | A | A | A | UE | UE |

**Anmerkung:**
[1)] Nur ManagerRoleFunctionalUnits sind erlaubt.
[2)] Nur AgentRoleFunctionalUnits sind erlaubt.
[3)] ManagerRoleFunctionalUnits, AgentRoleFunctionalUnits, oder beide dürfen anwesend sein.
[4)] Agent kann nur mit AgentRoleFunctionalUnits, die eine Untermenge der ManagerRoleFunctionalUnits des Initiators sind, antworten.
[5)] Manager kann nur mit ManagerRoleFunctionalUnits, die eine Untermenge der AgentRoleFunctionalUnits des Initiators sind, antworten.
[6)] Antworten mit ManagerRoleFunctionalUnits als Untermenge der AgentRoleFunctionalUnits des Initiators, und mit AgentRoleFunctionalUnits als Untermenge der ManagerRoleFunctionalUnits des Initiators.

**Tab. 1. Verhandlungsregeln der Funktionseinheiten:** Die in Tabelle verwendeten Abkürzungen haben die folgenden Bedeutungen:

- UserInfo: stellt die Spezifikationen für eine Verbindung bezüglich der optionalen System-Management-Funktionseinheiten dar.
- VV: Verhandelte Verbindung; der Kontext erlaubt keine Operationen und Meldungen außerhalb des ausgehandelten Kontextes. Dies ist durch die gesetzten Bits in UserInfo des Beantworters erkennbar. Das heißt, daß der Beantworter die gesetzten Bits nicht ändern oder zusätzliche Funktionseinheiten oder Funktionseinheit-Pakete vorschlagen wird.
- UE: Undefinierte Einschränkungen; der Kontext definiert oder beschränkt während dieser Verbindung keine Meldungen oder Operationen. Andere mögliche Einschränkungen von Meldungen und Operationen sind nicht Gegenstand dieses Standards.
- A: Abbruch; tritt diese Bedingung auf, so kann der Initiator die Verbindung abbrechen.

Die Errichtung des Management-Wissens ist möglich, bevor irgendwelche Kommunikation stattfindet, also während des Verbindungsaufbaus und während des Betriebs nach dem Verbindungsaufbau. Nachdem eine Verbindung zum Zweck des System-

Managements aufgebaut worden ist, darf ein Mechanismus zur Modifikation des Management-Wissens benutzt werden. Abhängig von den Anforderungen der Funktionen können verschiedene Kommunikations-Dienste angeboten werden, wie File-orientierte Management-Operationen oder einfaches Anfrage/Antwort-Protokoll.

#### 2.5.1.5 Organisations-Aspekte

Die Organisations-Aspekte dieses Modells beschreiben die Aspekte, die das OSI-Netzwerkmanagement als ein verteiltes Management repräsentieren. Das Manager-Agent-Konzept, das bei den Kommunikations-Aspekten beschrieben wurde, ist ein Beispiel, das den verteilten Charakter des OSI-Netzwerkmanagements verdeutlicht und der Organisations-Strategie des OSI-Netzwerkmanagements zuzuordnen ist. Um das Ziel des Netzwerkmanagements, die effektive Verwaltung der verwalteten Objekte, zu erreichen, wird durch OSI das Konzept der Management-Bereiche *(management domains)* eingeführt. Unter einem Management-Bereich versteht man einen Satz von Ressourcen zur Erfüllung bestimmter Organisations-Forderungen. Diese Ressourcen werden für die System-Management-Zwecke durch verwaltete Objekte dargestellt. Ein Management-Bereich besitzt einen Namen zur eindeutigen Identifikation des Bereiches, identifiziert die den Bereich repräsentierenden verwalteten Objekte und stellt die Beziehungen zwischen den Bereichen fest. Die OSI-Netzwerkmanagement-Umgebung kann für die Funktionszwecke, wie z.B. für das Sicherheit-, Abrechnung- oder Fehler-Management in Bereiche eingeteilt werden. Die Einteilung der OSI-Netzwerkmanagement-Umgebung kann auch geographische, technologische oder organisatorische Gründe haben. Dieses Konzept erfüllt die Forderungen nach temporärer Zuordnung und Modifizierung der Rollen der Manager und Agents innerhalb eines Bereiches, und auch die Ausübung der Steuerung in einer konsistenten Art.

Zusätzlich zu den Organisations-Forderungen existieren die Verwaltungs-Forderungen, wie die Errichtung und Aufrechterhaltung der jeweiligen Autoritäten jedes Management-Bereiches, die Organisation der Überlappung der Bereiche oder die Überwachung der Übertragung von Steuerungsinformationen für die Ressourcen zwischen den Management-Bereichen. Diese Forderungen werden durch die Definition eines speziellen Management-Bereiches erfüllt. Dieser Management-Bereich wird Management-Verwaltungs-Bereich *(Management administrative domain)* genannt. Unter einem Management-Verwaltungs-Bereich wird ein Management-Bereich verstanden, dessen verwaltete Objekte unter der Verantwortung nur einer Verwaltungsautorität stehen. Das Konzept der Management-Bereiche ist nicht Gegenstand der Standardisierung, obwohl die Wechselwirkung zwischen den Bereichen der Normung unterworfen sein kann.

### 2.5.2 (N)-Schicht-Management

Das (N)-Schicht-Management *((N)-layer management)* liefert Mechanismen zur Überwachung, Steuerung und Koordination von verwalteten Objekten, die in Verbindung mit den Kommunikationsaktivitäten innerhalb der (N)-Schicht stehen, also den verwalteten (N)-Schicht-Objekten [ISO7498-4]. Hierzu werden die (N)-Schicht-Protokolle verwendet, die durch die darunterliegenden Protokolle unterstützt werden. Sie können keine Kommunikations-Fähigkeiten bereitstellen, die die darüberliegenden Schichten anbieten. Die innerhalb der (N)-Schicht-Protokolle übertragenen Informationen können

nur zwischen den (N)-Schicht-Partner-Instanzen und über die (N)-Subsysteme, in denen die Instanzen lokalisiert sind, ausgetauscht werden. Die (N)-Schicht-Protokolle sollen verwendet werden, wenn die Benutzung der System-Management-Protokolle unangemessen ist, oder die System-Management-Protokolle nicht erreichbar sind.
Die (N)-Schicht-Management-Protokolle liefern Funktionen, wie z.B. die Kommunikation der Parameterwerte beim Betrieb der verwalteten (N)-Schicht-Objekte, die Prüfung der durch die (N-1)-Schicht gelieferten Funktionalität, die Übertragung von Fehler-Informationen und die Beschreibung und Diagnose der Fehler, die mit dem Betrieb der (N)-Schicht verbunden sind. Jedes (N)-Schicht-Management-Protokoll ist von den anderen Schicht-Protokollen unabhängig. Das im Rahmen der Transportschicht definierte *network connection management subprotocol* [ISO8073AD1] ist ein Beispiel für ein (N)-Schicht-Protokoll.

Das (N)-Schicht-Management befaßt sich nicht nur mit einer einzelnen Instanz, sondern mit allen Instanzen der Kommunikation auf der (N)-Schicht. Im Gegensatz zum System-Management hat das (N)-Schicht-Management begrenzten Wirkungs- und Gültigkeits-Bereich, der sich nur auf die (N)-Schicht bezieht. Die (N)-Schicht darf durch die Benutzung der Protokolle des System-Managements oder durch die Benutzung der Protokolle des (N)-Schicht-Managements überwacht werden. Die Semantik der (N)-Schicht-Management-Informationen und -Operationen müssen mit solchen von System-Management konsistent bleiben. Das Verändern der die (N)-Schicht betreffenden Parameter ist ein Beispiel zur Verdeutlichung dieses Gültigkeitsbereiches. Schließlich ist ein verwaltetes (N)-Schicht-Objekt ein verwaltetes Objekt, das für eine individuelle Schicht spezifisch ist.

### 2.5.3 (N)-Schicht-Operation

Die (N)-Schicht-Operation *((N)-layer operation)* stellt Mechanismen bereit, die der Überwachung und Steuerung einer einzelnen Instanz während der Kommunikation dienen. Zur Kommunikation werden die normalen (N)-Protokolle verwendet. Die durch (N)-Protokolle übertragenen Informationen müssen sich von den für andere Zwecke übertragenen Informationen unterscheiden. Hierfür sind die (N)-Protokolle zuständig. Die durch diese Protokolle übertragenen Management-Informationen sind unter anderem

- Parameter, welche beim Verbindungsaufbau die aktuelle Verbindung kennzeichnen (wie die Flußkontroll-Fenstergröße der Vermittlungsschicht beim X.25-PLP oder die Auswahl eines Kontextes für die Darstellungsschicht),
- Parameter, welche beim Verbindungsabbau Informationen über die frei gewordenen Instanzen beinhalten,
- Parameter, welche die Umgebung modifizieren können, in der diese Instanz der Kommunikation operiert,
- Informationen über Fehlerursachen (z.B. Ursache- und Diagnose-Kodes in X.25-PLP RESET-Paketen) und
- Abrechnungs-Information über eine aktive Verbindung etc. [Rose89, ISO7498-4].

Die (N)-Schicht-Management-Instanzen und (N)-Instanzen operieren auf verwalteten Objekten unabhängig voneinander. Die (N)-Schicht-Protokolle und die (N)-Protokolle unterscheiden sich durch die Benutzung von (N-1)-Schicht-Adressierungsmechanismen oder die Unterscheidungsmechanismen innerhalb der (N)-Schicht.

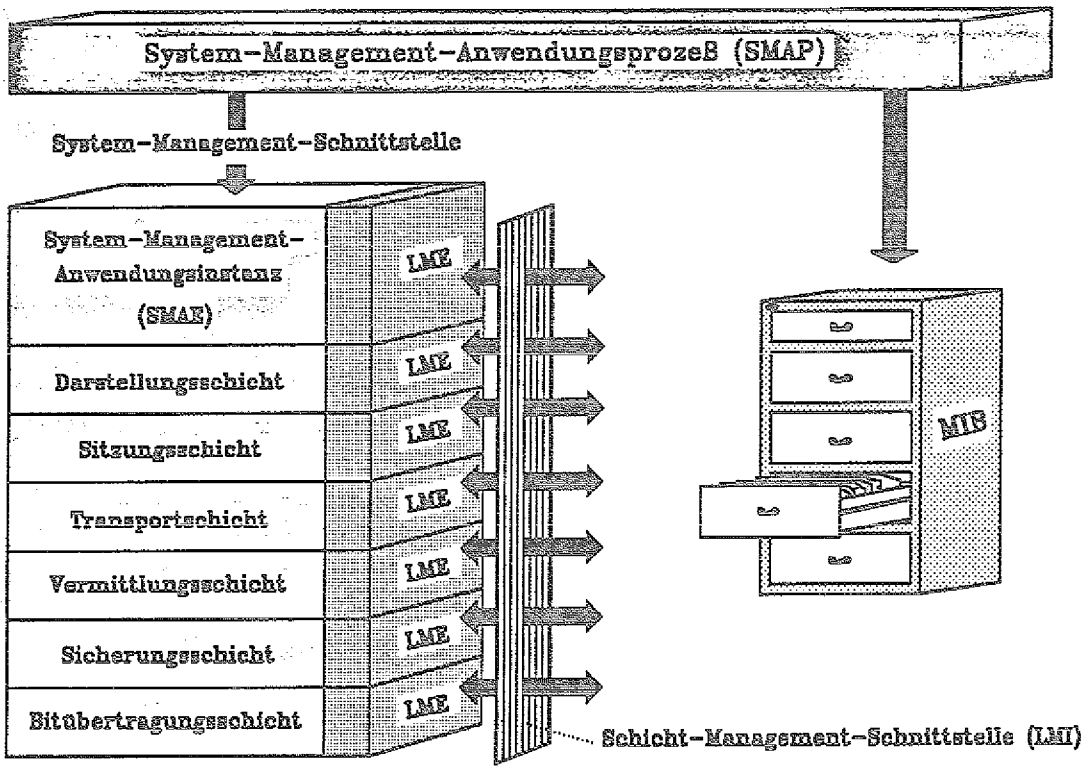



**Abb. 12. OSI-Netzwerkmanagement-Architektur:** Die OSI behandelt die Managementaufgabe durch eine auf der siebten Schicht angesiedelte Anwendung. Der SMAP erbringt die erforderlichen Funktionen im Rahmen des Manager-Agent-Konzepts, während die SMAE für die Kommunikation zuständig ist. Jede Schicht benutzt die Schicht-Management-Instanz *(layer management entity, LME)*, um Zugriff auf die Ansammlung von Management-Informationen, Management Information Base (MIB), zu haben.

### 2.5.4 Management-Informationsbasis

Die verwalteten Objekte innerhalb eines offenen Systems bilden zusammen mit ihren Attributen die Management-Informationsbasis *(Management Information Base, MIB)* des Systems, wobei nur die verwalteten Objekte innerhalb der OSI-Umgebung der Standardisierung unterworfen sind. Die MIB ist rein konzeptionell ein Ort zur Aufbewahrung von Informationen über das Netz und sollte nicht mit einer Datenbank gleichgesetzt werden [Rose89]. Sie beinhaltet die Informationen, die durch OSI-Netzwerkmanagement-Protokolle übertragen und beeinflußt werden dürfen. Die in der MIB vorhandenen Informationen stammen erstens von den lokalen Agenten und zweitens aus den entfernten offenen Systemen. Sie werden durch System-Management-Protokolle, (N)-Schicht-Protokolle und (N)-Protokolle bereitgestellt.

Nur die abstrakte Syntax und Semantik der in einer MIB vorhandenen Management-Informationen, d.h. die logische Struktur der Informationen werden normiert, da sie für die Definition der Management-Protokolle festgelegt sein müssen. Die MIB unterliegt keiner Einschränkung bezüglich der Form der physikalischen Speicherung und der in der MIB verwendeten Datenstrukturen (logische Speicherung). Es existiert keine Festlegung bezüglich der Vorverarbeitung der Daten in der MIB. Die System-Management-, Schicht-Management- und Schicht-Instanzen können gemeinsam die MIB benutzen und modifizieren. Zur Modifizierung werden Management-Protokolle verwendet. Die vom System-Management bereitgestellten Dienste werden in Anspruch genommen, wenn Prozesse der Anwendungsschicht auf die MIB zugreifen (vgl. Abb. 12).

### 2.5.5 Dienst und Protokoll im OSI-Netzwerkmanagement

Der allgemeine Management-Informationsdienst *(Common Management Information Service, CMIS)* und das allgemeine Management-Informationsprotokoll *(Common Management Information Protocol, CMIP)* sind im Rahmen des OSI-Netzwerkmanagements definiert worden, um dem System-Management Dienste und Protokoll zur Verfügung zu stellen. Sie dienen der Übertragung von [SlTJ91]:

- Operationen auf die verwalteten Objekte,
- Meldungen von den verwalteten Objekten,
- Ergebnissen von den Operationen,
- Ergebnissen von den Meldungen und
- Fehlern.

#### 2.5.5.1 Allgemeiner Management-Informationsdienst

Der CMIS ist in der Allgemeinen Management-Informationsdienst-Definition *(Common Management Information Service Definition)* [ISO9595] definiert worden. Die Definition umfaßt einen Satz von Dienstelementen und ihren Parametern, wie auch die Definition der notwendigen Informationen für die Beschreibung der Semantik jedes Dienstelements. Ein Anwendungsprozeß benutzt die Management-Informationsdienste, um Informationen und Befehle zum Zweck des System-Managements austauschen zu können.

Für die Steuerung der Anwendungsverbindungen werden die Dienste benutzt, die in der Dienstdefinition für Verbindungs-Steuerungsdienstelement *(service definition for the association control service element)* [ISO8649] beschrieben sind, d.h. der CMIS definiert keine speziellen Dienste zur Errichtung und Freigabe einer Verbindung. Die Verbindungsdienste werden durch ACSE zur Verfügung gestellt und zur Verhandlung über die Funktionseinheiten und Protokollversionen verwendet:

- Der A-ASSOCIATE-Dienst wird benutzt, um eine Management-Verbindung zu errichten. Dieser Dienst wird im Bestätigt-Modus *(confirmed)* angewendet und besitzt die Parameter, die den Anwendungs-Kontext, die Darstellungs- und die Sitzungs-Forderungen kennzeichnen.

- Der A-RELEASE Dienst ist für die normale Beendigung einer Verbindung vorgesehen, wobei der Verlust von Informationen vermieden wird. Dieser Dienst wird im Bestätigt-Modus angefordert.

- Der A-ABORT-Dienst sorgt für die abrupte Unterbrechung der Verbindung durch ACSE-Dienstanbieter, wobei Informationen verloren gehen können. Dieser Dienst wird im Unbestätigt-Modus *(non-confirmed)* benutzt.

Zur Informationsübertragung sind zwei Diensttypen, der Management-Meldungsdienst *(management notification service)* und der Management-Operationsdienst *(management operation service)*, definiert worden. Die Dienste werden durch *common management information service element (CMISE)*, ein spezielles Anwendungsdienstelement (ASE), zur Verfügung gestellt.

Der ***Management-Meldungsdienst*** liefert einen allgemeinen Dienst, der zur Übertragung der Management-Informationen für die Meldungen, benutzt werden kann [SlTJ91]:

- Der M-EVENT-REPORT-Dienst wird durch den CMISE-Dienstbenutzer aufgerufen, um den Partner-CMISE-Dienstbenutzer über das Auftreten eines Ereignisses bei einem verwalteten Objekt zu benachrichtigen. Sind dem CMISE-Dienstbenutzer Informationen über das Ereignis bekannt, so werden diese Informationen dem Aufforderung-Dienstelement für M-EVENT-REPORT als Argumente beigefügt. Die übertragenen Informationen sind unter anderem die Objektklassen, die Ereignistypen und die optionalen, ereignisspezifischen Informationen. Die Meldungsspezifikationen eines verwalteten Objekts definieren die Bedeutung der durch diesen Dienst übertragenen Informationen. Dieser Dienst kann im Bestätigt- oder Unbestätigt-Modus angefordert werden. Eine Antwort wird erwartet, wenn die bestätigte Form des Dienstes benutzt wird.

Der ***Management-Operationsdienst*** umfaßt allgemeine Dienste, die zur Übertragung der Management-Informationen für die Operationen benutzt werden. Die folgenden Dienste werden durch einen CMISE-Dienstbenutzer aufgerufen, um den Partner-CMISE-Dienstbenutzer aufzufordern, die jeweilige Aufgabe zu erbringen. Der Partner-CMISE-Dienstbenutzer überprüft das empfangene Hinweis-Dienstelement nach bestimmten Parametern. Ist die Richtigkeit der Semantik der Parameter gewährleistet, so wird in der Regel die Aufgabe erbracht.

- Der M-GET-Dienst ist für die Gewinnung der Management-Informationen vorgesehen. Dabei werden solche Informationen geholt, wie ein oder mehrere Attributwerte von einem oder mehreren verwalteten Objekten.

- Der M-SET-Dienst wird zur Modifikation der Eigenschaften eines verwalteten Objekts benutzt. Dieser Dienst ermöglicht, einen oder mehrere Attribut-Werte eines oder mehrerer verwalteten Objekte zu ändern.

- Der M-ACTION-Dienst ist zur Ausführung verschiedener Aktionen auf verwalteten Objekten vorgesehen. Dieser Dienst überträgt unter anderem den Aktionstyp und die optionalen aktionsspezifischen Informationen. Die Bedeutung der durch diesen Dienst übertragenen Informationen ist von den Aktionsspezifikationen der verwalteten Objekte abhängig.

- Der M-DELETE-Dienst ermöglicht das Entfernen einer verwalteten Objekt-Instanz.

- Der M-CREATE-Dienst bewirkt die Schaffung der Darstellung einer verwalteten Objekt-Instanz zusammen mit den dazugehörigen Werten der Management-Information.

- Der M-CANCEL-GET-Dienst wird benutzt, um einen vorher angeforderten M-GET-Dienst abzusetzen.

Die Informationen über den Dienst-Modus, bestätigt oder unbestätigt *(confirmed or non-confirmed)*, sind in der Tab. 2 dargestellt. Außerdem wird verdeutlicht, durch welche Dienste die in der Struktur von Management-Information *(Structure of Management Information, SMI)* [ISO10165] definierten Operationen modelliert werden.

| Dienst | Typ | SMI-Operation |
|---|---|---|
| M-CANCEL-GET | bestätigt | - |
| M-EVENT-REPORT | bestätigt/unbestätigt | - |
| M-GET | bestätigt | Get attribute value |
| M-SET | bestätigt/unbestätigt | Replace attribute value<br>Set-to-default value<br>Add member<br>Remove member |
| M-ACTION | bestätigt/unbestätigt | Action |
| M-CREATE | bestätigt | Create |
| M-DELETE | bestätigt | Delete |

**Tab. 2. Die CMISE-Dienste:** Die CMISE-Dienste werden durch CMISE-Dienstbenutzer aufgerufen, damit der Partner-CMISE-Dienstbenutzer den jeweiligen Dienst erbringt. Der Modus-Typ ist für jeden Dienst angegeben. Ist ein Dienst im Bestätigt-Modus angefordert, so wird auf eine Antwort gewartet. Die CMISE-Dienste modellieren die in SMI definierten Operationen.

Die CMIS-Dienste basieren auf der einfachen Anfrage/Antwort-Methode. Die anderen charakteristischen Eigenschaften vom CMIS werden im folgenden beschrieben, wie Bereichsbestimmung *(scoping)* und Filterung *(filtering)*, Synchronisation *(synchronization)*, verbundene Antworten *(linked replies)* und die Funktionseinheiten. Es ist zu berücksichtigen, daß die Operationsdienste M-GET, M-SET, M-ACTION und M-DELETE die Fähigkeit besitzen, die CMIS-Eigenschaften Bereichsbestimmung, Filterung, Synchronisation und verbundene Antworten zu benutzen.

Die Namen der verwalteten Objektinstanzen werden hierarchisch in einem Management-Informationsbaum festgehalten. Die Management-Informationen können als eine Ansammlung von verwalteten Objekten betrachtet werden, welche Attribute besitzen und Ereignisse *(events)* und Aktionen *(actions)* definiert haben. Um auf den Objekten Operationen auszuführen, müssen sie zuerst ausgewählt werden. Der Auswahlvorgang der verwalteten Objekte innerhalb eines Enthaltungs-Baumes kann in zwei Phasen, Bereichsbestimmung und Filterung, verlaufen.

Unter ***Bereichsbestimmung*** *(scoping)* versteht man die Identifikation der verwalteten Objekte relativ zu einem verwalteten Basisobjekt, aus denen dann durch Filterung Objekte ausgewählt werden können. Die bei der Bereichsbestimmung identifizierten verwalteten Objekte können, bei einem Unterbaum eines Enthaltungsbaumes mit dem ver-

walteten Basisobjekt als Wurzel des Unterbaumes, die folgenden Elemente des Baumes, wie in der Abb. 13 dargestellt, spezifizieren:

- nur das verwaltete Basisobjekt,
- nur die n-te Stufe des Baumes,
- das verwaltete Basisobjekt mit allen Elementen des Unterbaumes einschließlich der n-ten Stufe und
- den gesamten Unterbaum.

Unter ***Filterung*** *(filtering)* ist die Auswahl der verwalteten Objekte nach bestimmten Kriterien aus den Objekten, die schon durch Bereichsbestimmung ausgewählt worden sind, zu verstehen. Unter den Kriterien, die die Filterung ermöglichen, wird die Prüfung der Attributwerte der verwalteten Objekte verstanden, wie z.B. die Prüfung nach dem Vorhandensein und dem Vergleich der Attribute. Diese Prüfungen können auch komplexer werden, indem sie durch die logischen Operatoren kombiniert werden.

Unter ***Synchronisation*** *(synchronization)* versteht man die Ausübung von Operationen auf vielfachen Objekten, die schon durch Bereichsbestimmung oder Filterung ausgewählt worden sind. Der Synchronisationsparameter erlaubt dem CMIS-Dienstbenutzer, die Art der Synchronisation der Operationen über die verwalteten Objektinstanzen zu bestimmen. Der Parameter kann die zwei Synchronisationstypen, unteilbar *(atomic)* oder teilbar *(best effort)*, kennzeichnen [ISO9595]. Die unteilbare Synchronisation spezifiziert die Synchronisation derart, daß alle oder keine verwalteten Objekte einer Operation unterworfen werden. Im Gegensatz dazu bezeichnet die teilbare Synchronisation eine Synchronisationsart, die eine erreichbare Teilmenge der verwalteten Objekte berücksichtigt. Der CMIS stellt keinen Parameter zur synchronisierten Behandlung der Attribute eines verwalteten Objekts zur Verfügung. Ein solcher Parameter soll als Teil des Verhaltens eines verwalteten Objekts definiert werden.

Unter ***verbundenen Antworten*** *(linked replies)* versteht man die vielfachen Antworten, die zu einer einzelnen Operationsanforderung bereitgestellt werden. Diese Eigenschaft kann nur dann verwendet werden, wenn die Bereichsbestimmung oder die Filterung schon benutzt worden ist. Die verbundenen GET-Antworten können durch CANCEL-GET abrupt beendet werden. Eine einzelne Management-Operation kann vielfache Antworten auslösen, wenn der aufrufende CMISE-Dienstbenutzer vielfache verwaltete Objekte auswählt, oder der angeforderte M-ACTION-Dienst für ein einzelnes verwaltetes Objekt eine Aktion ausführt, die nach ihrer Definition mit vielfachen Antworten reagiert.

Unter den ***Funktionseinheiten*** *(functional units)* sind die allgemeinen Dienste zu verstehen, die die Identifikation der Systemfähigkeiten beim Verbindungsaufbau erlauben. Die Kernfunktionseinheit *(kernel functional unit)* definiert den minimalen Funktionensatz vom CMIP und beinhaltet alle CMISE-Dienste in der Tab. 2 außer dem M-CANCEL-GET.

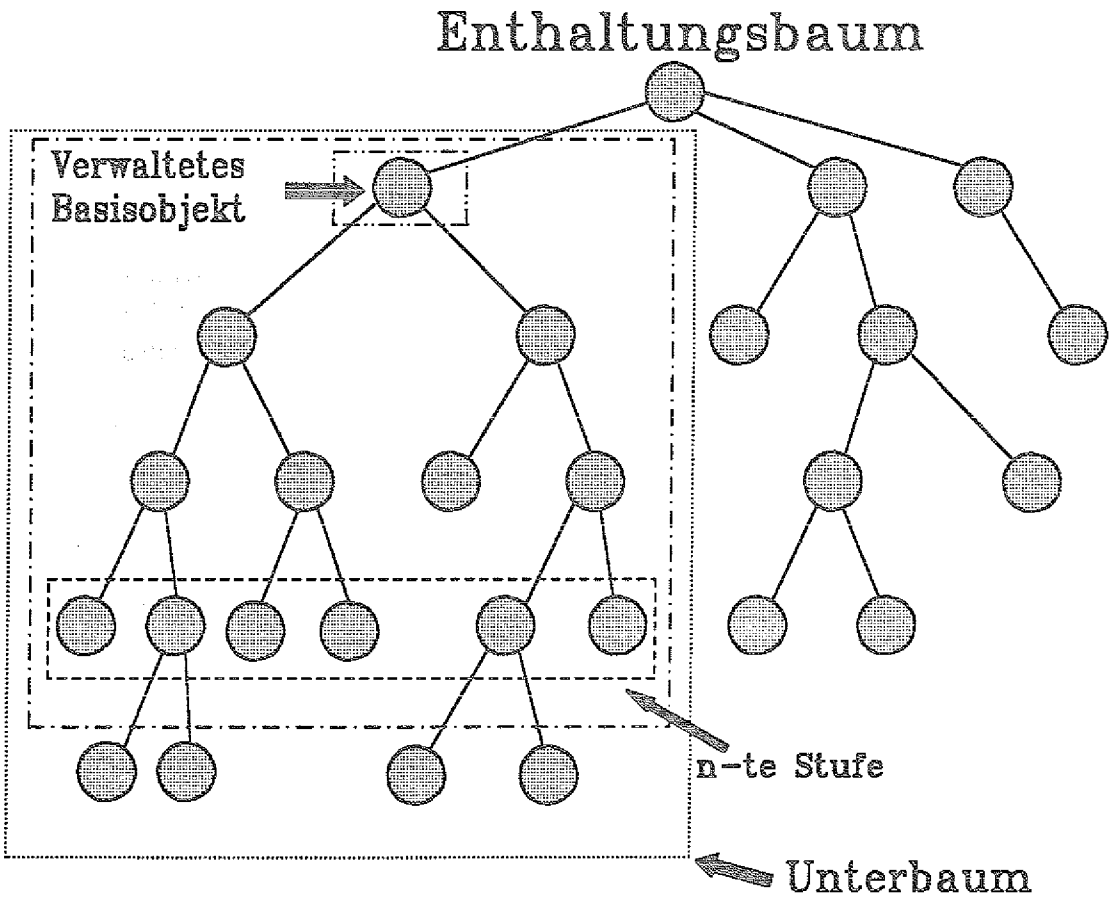



**Abb. 13. Bereichsbestimmung und Filterung:** Bei der Bereichsbestimmung werden die Elemente bezüglich eines verwalteten Basisobjekts ausgewählt. Die Auswahlmöglichkeiten sind in der Abbildung durch verschiedene Linien-Typen veranschaulicht. Die Filterung erlaubt die Identifikation der Objekte nach bestimmten Kriterien aus den schon durch Bereichsbestimmung ausgewählten verwalteten Objekten.

Die zusätzlichen Funktionseinheiten erlauben den Diensten in der Kernfunktionseinheit bestimmte Parameter zu benutzen [ISO9595]:

- Vielfachobjektauswahl-Funktionseinheit *(multiple object selection functional unit)* stellt den Diensten in der Kernfunktionseinheit den Bereichsbestimmungs- und Synchronisationsparameter zur Verfügung. Diese Parameter sind im M-EVENT-REPORT und M-CREATE nicht vorhanden.

- Filter-Funktionseinheit *(filter functional unit)* macht den Diensten in der Kernfunktionseinheit den Filterparameter verfügbar. Die Dienste M-EVENT-REPORT und M-CREATE besitzen keinen Filterparameter.

- Vielfachantwort-Funktionseinheit *(multiple reply functional unit)* ermöglicht der Kernfunktionseinheit, den Parameter Verbundene Identifikation *(linked identification)* zu benutzen. Der Parameter Verbundene Identifikation ist nicht in den M-EVENT-REPORT und M-CREATE Diensten vorhanden.

- Zusatzdienst-Funktionseinheit *(extended service functional unit)* stellt zusätzlich zum P-DATA-Dienst andere Darstellungsschichtdienste bereit.

- Aufhebungs-Get-Funktionseinheit erlaubt die Benutzung des M-CANCEL-GET-Dienstes.

#### 2.5.5.2 Allgemeines Management-Informationsprotokoll

Das CMIP ist in der Allgemeinen Management-Informationsprotokoll-Spezifikation *Common Management Information Protocol Specification* [ISO9596] definiert worden. Das Protokoll dient den Anwendungsschichtinstanzen zum Austausch von Management-Informationen. Dieses Protokoll spezifiziert die Protokollelemente, die für die Bereitstellung der in [ISO9595] beschriebenen Operations- und Meldungsdienste benutzt werden können. Die vom CMIP unterstützten Dienste sind in der Tab. 2 auf Seite 41 dargestellt. Die dem CMIP zugrundeliegenden Dienste sind unter anderem die in [ISO9072-1] definierten ROSE-Dienste:

- RO-INVOKE: Dieser Dienst ermöglicht, daß eine aufrufende Anwendungsinstanz die Durchführung einer Operation auffordert, die dann von einer ausführenden Anwendungsinstanz erbracht wird.

- RO-RESULT: Eine ausführende Anwendungsinstanz benutzt diesen Dienst, um die positive Antwort einer erfolgreich ausgeführten Operation der aufrufenden Anwendungsinstanz mitzuteilen.

- RO-ERROR: Durch diesen Dienst hat eine ausführende Anwendungsinstanz die Möglichkeit, die Antwort auf eine nicht erfolgreich ausgeführte Operation der aufrufenden Anwendungsinstanz mitzuteilen.

- RO-REJECT-U: Eine Anwendungsinstanz ist durch diesen Dienst in der Lage, eine Aufforderung oder die Antwort einer anderen Anwendungsinstanz zurückzuweisen, wenn der ROSE-Benutzer ein Problem entdeckt hat. Unter einem ROSE-Benutzer versteht man eine anwendungsspezifische Funktion, die die Abbildung der OPERATION- und ERROR-Macros der RO-Notation auf ROSE ermöglicht. Die RO-Notation ist die Notation, die für die Spezifikation der Fernoperationen benutzt wird [ISO9072-1, ISO9072-2].

Die oben genannten ROSE-Dienste werden im Unbestätigt-Modus aufgefordert. Das ROSE benutzt seinerseits den Darstellungsdienst, der im [ISO8822] definiert ist. Der Darstellungsdienst stellt die Hilfsmittel zur Definition des Darstellungskontextes bereit. Der Darstellungskontext stellt die Forderungen seiner Benutzer nach Informationsübertragung fest. Ist die Zusatzdienst-Funktionseinheit[9] erfolgreich ausgehandelt worden, so können ROSE-APDUs zusätzlich zum P-DATA-Dienst den anderen Darstellungsschichtdiensten angepaßt werden, d.h. andere Darstellungsschichtdienste stehen über die Zusatzdienst-Funktionseinheit zur Verfügung. Aufgrund der Abhängigkeit des CMIP vom ROSE besitzt CMIP keine Zustandstabellen, Ereignislisten, Vorhersage- und Aktions-Tabellen [Blac91].

---

9 siehe „2.5.5.1 Allgemeiner Management-Informationsdienst“ auf Seite 39

Das CMIP setzt auch die Verwendung der ACSE-Dienste, A-ASSOCIATE, A-RELEASE und A-ABORT voraus [SITJ91]. Sie wurden im „2.5.5.1 Allgemeiner Management-Informationsdienst“ auf Seite 39 beschrieben. Die Syntax der ROSE-Anwendungsprotokoll-Dateneinheiten, die der Übertragung von CMIP-Informationen dienen, werden unten zur Veranschaulichung angegeben.

```
ROSEapdus ::= CHOICE { roiv-apdu   [1] IMPLICIT ROIVapdu,
                       rors-apdu   [2] IMPLICIT RORSapdu,
                       roer-apdu   [3] IMPLICIT ROERapdu,
                       rorj-apdu   [4] IMPLICIT RORJapdu }

ROIVapdu ::= SEQUENCE { invokeID          InvokeIDType,
                        linked-ID         [0] IMPLICIT InvokeIDType OPTIONAL,
                        operation-value   OPERATION,
                        argument          ANY DEFINED BY operation-value OPTIONAL }

RORSapdu ::= SEQUENCE { invokeID          InvokeIDType,
                        SEQUENCE { operation-value     OPERATION,
                                   result              ANY DEFINED BY opration-value } OPTIONAL }

ROERapdu ::= SEQUENCE { invokeID          InvokeIDType,
                        error-value       ERROR,
                        parameter         ANY DEFINED BY error-value OPTIONAL }

RORJapdu ::= SEQUENCE { invokeID     CHOICE { InvokeIDType,
                                              NULL },
                        problem      CHOICE { [0] IMPLICIT GeneralProblem,
                                              [1] IMPLICIT InvokeProblem,
                                              [2] IMPLICIT ReturnResultProblem,
                                              [3] IMPLICIT ReturnErrorProblem } }
```

Die Prozeduren zur Übertragung von Management-Informationen zwischen Anwendungsinstanzen sind in [ISO9596] spezifiziert worden. Sie definieren die Übertragung von CMIP-PDUs und deren Struktur, Kodierung und Beziehung zu den CMIS-Dienstelementen. Das CMIP ist, wie die anderen OSI-Protokolle, für die Zusammensetzung und den Austausch von PDUs Regeln unterworfen. „Alle CMIP-PDUs werden durch ASN.1 definiert“[Blac91].

Die Prozeduren bestimmen, wie verschiedene Felder in der CMIP-PDU zu verstehen sind. Sie legen nicht fest, was der aufrufende CMISE-Dienstbenutzer mit den angeforderten Informationen machen soll oder wie der ausführende CMISE-Dienstbenutzer den Aufruf bearbeiten soll. Die CMIP-PDUs werden durch die CMIP-Maschine *(CMIPM)* erzeugt, nachdem sie ein CMIS-Aufforderung- oder CMIS-Beantwortungs-Dienstelement empfangen hat. Die CMIPM empfängt jede richtig geformte *(well-formed)* CMIP-PDU und leitet sie dann an den ausführenden CMISE-Dienstbenutzer weiter. Ist aber ein CMIP-PDU falsch geformt oder enthält sie keine Meldung oder Operation, so wird eine PDU durch die CMIPM zurückgeschickt, die auf die Ablehnung der vorher empfangenen PDU hinweist. Die CMIPM leitet die CMIP-PDUs durch die CMIS-Hinweis- oder CMIS-Bestätigung-Dienstelemente an die CMISE-Dienstbenutzer weiter.

Zur Beschreibung der Prozeduren zur Übertragung von Management-Informationen wird hier als Beispiel die GET-Prozedur gewählt [ISO9596]. Die Syntax der Opera-

tionen M-GET und M-Linked-Reply werden zur Veranschaulichung als Beispiel dargestellt [ISO9596].

```
m-Get OPERATION
  ARGUMENT      GetArgument
  RESULT        GetResult
  ERRORS {      accessDenied,
                classInstanceConflict,
                complexityLimitation,
                getListError,
                invalidFilter,
                invalidScope,
                noSuchObjectClass,
                noSuchObjectInstance,
                processingFailure,
                syncNotSupported
        }
  LINKED {      m-Linked-Reply }
  ::= localValue 3

m-Linked-Reply OPERATION
  ARGUMENT      LinkedReplyArgument
  ::= localValue  2
```

Die GET-Prozedur, wie alle anderen Prozeduren, durchläuft die vier Phasen, die jeweils

- Aufruf,
- Empfang,
- Antwort und
- Empfang der Antwort

genannt werden. In der ***Aufrufphase*** wird die GET-Prozedur durch ein M-GET-Aufforderung-Dienstelement initialisiert. Die CMIPM empfängt das Aufforderung-Dienstelement und konstruiert eine APDU, die die M-GET Operation auffordert. Diese APDU wird dann durch Zuhilfenahme der RO-INVOKE-Prozedur, eines ROSE-Dienstes, zur CMIPM auf der Empfangsseite geschickt. Die erste Phase ist damit beendet.

Die ***Empfangsphase*** beginnt mit dem Empfang einer APDU, die eine M-GET-Operation auffordert. Ist die APDU annehmbar *(well-formed)*, so wird ein M-GET-Hinweis-Dienstelement für den ausführenden CMIS-Dienst-Benutzer erzeugt. Ist sie nicht annehmbar, so wird eine APDU, die eine Fehlermeldung beinhaltet, zurückgeschickt. Dabei wird die RO-REJECT-Prozedur, ein ROSE-Dienst, benutzt. Die Empfangsphase ist damit beendet.

Die ***Antwortphase*** beschäftigt sich mit dem Verhalten der CMIPM auf der Empfangsseite. Die CMIPM akzeptiert die M-GET-Beantwortungs-Dienstelemente nur dann, wenn das letzte M-GET-Beantwortungs-Dienstelement den Parameter Verbundene Identifikation *(linked-ID)* nicht besitzt. Davor können keine oder mehrere M-GET-Beantwortungs-Dienstelemente mit dem Parameter Verbundene Identifikation erscheinen. Für jedes M-GET-Beantwortungs-Dienstelement mit dem Parameter Verbundene Identifikation konstruiert die CMIPM eine APDU, die zur Aufforderung der M-Linked-Reply-Operation dient. Jede APDU wird dann mittels der RO-INVOKE-

Prozedur übertragen. Die CMIPM konstruiert für das M-GET-Beantwortungs-Dienstelement ohne den Parameter Verbundene Identifikation eine APDU, die die M-GET-Operation bestätigt. Danach wird eine APDU mit Hilfe der RO-RESULT-Prozedur geschickt, wenn die in den Dienstelementen vorhandenen Parameter auf eine korrekt durchgeführten Operation hinweisen. Ist die Operation aber nur teilweise korrekt oder aufgrund eines Fehlers nicht ausgeführt worden, so wird eine APDU mittels RO-ERROR übertragen. Hiermit wird die Antwortphase abgeschlossen.

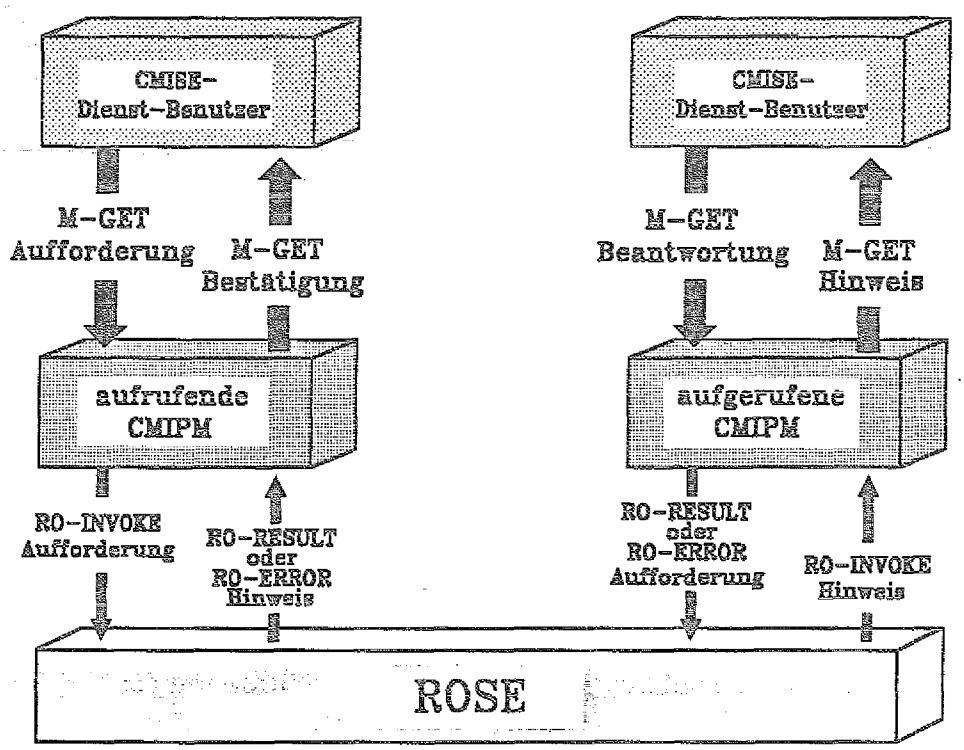



**Abb. 14. Die CMIP-Operation M-GET:** Die Abbildung stellt dar, wie die M-GET-Operation zur Gewinnung von Informationen realisiert wird. Dabei werden die M-GET-Dienstelemente zur Übertragung von Informationen zwischen dem CMISE-Dienstbenutzer und den Protokollmaschinen (CMIPMs) benutzt. Die CMIPMs nehmen die ROSE-Dienste zum Austausch der APDUs in Anspruch. In der Abbildung wurde vorausgesetzt, daß die von den CMIPMs empfangenen APDUs annehmbar *(well-formed)* sind. Deshalb wurden die RO-REJECT-U Dienstelemente nicht eingezeichnet.

Zur Veranschaulichung wird die Syntax der CMIP benutzten ROIVapdu und RORSapdu für die M-Get-Operation angegeben.

```
m-Get OPERATION ::= localValue 3
ROIV-m-Get ::= ROIVapdu (WITH COMPONENTS
                    { invokeID          PRESENT,
                      linked-ID         ABSENT,
                      operation-value   (m-Get),
                      argument          (INCLUDES GetArgument } )
```

```
RORS-m-Get ::= RORSapdu (WITH COMPONENTS
                  { ... ,
                    invokeID          PRESENT,
                    --result sequence--(WITH COMPONENTS
                          { operation-value       (m-Get),
                            result                (INCLUDES GetResult)} )
                    --required only if there is a single reply to the ROIV-m-Get ROIVapdu
                      } )
```

Der *Empfang der Antwort*, die letzte Phase, beschreibt, wie die CMIPM auf der dienstaufrufenden Seite die empfangene Antwort behandelt. Zuerst kontrolliert die CMIPM, ob die APDUs annehmbar sind *(well-formed)*. Ist dies nicht der Fall, so wird eine APDU konstruiert, die eine Fehlermeldung enthält und unter Zuhilfenahme der RO-REJECT-U-Prozedur übertragen wird. Wenn aber die APDU annehmbar ist, wird ein M-GET-Bestätigung-Dienstelement für den aufrufenden CMISE-Dienstbenutzer bereitgestellt. Ist die empfangene APDU die letzte gewesen, d.h. eine APDU ohne den Parameter Verbundene Identifikation, so gilt die Prozedur danach als beendet.

### 2.5.6 Beispiele für System-Management-Funktionen

Die den SMFAs zur Verfügung stehenden Funktionen erfüllen die im OSI-Netzwerkmanagement den SMFAs zugeordneten Aufgaben. Um den allgemeinen Aufbau und Umfang der Funktionen zu verdeutlichen, werden beispielsweise im folgenden die Objekt-Management-Funktion, Zustand-Management-Funktion, Attribute für die Darstellung von Beziehungen und Alarmmeldungs-Funktion beschrieben.

#### 2.5.6.1 Objekt-Management-Funktion

Die Objekt-Management-Funktion ist in [ISO10164-1] definiert worden und beschreibt eine Anzahl von Diensten, die unter anderem das Erzeugen, Löschen und Ändern der Attributwerte von verwalteten Objekten ermöglichen. Daneben existiert ein anderer Dienst, der den verwalteten Objekten eine Namensänderung mitteilt. Um die für die verwalteten Objekte definierten Operationen und Meldungen auf die unterliegenden Kommunikationsdienste anwenden zu können, wurden innerhalb der Objekt-Management-Funktion Dienste definiert, die *Pass-Through Services* genannt werden. Diese Dienste definieren die Beziehung zwischen CMIS- und SMI-Operationen, wie in der Tab. 3 dargestellt ist [SITJ91].

Die Objekt-Management-Funktion definiert vier Meldungstypen:

- Object-Creation-Meldung: wird benutzt, wenn eine neue Objektinstanz erzeugt wird;
- Object-Deletion-Meldung: wird erzeugt, wenn eine Objektinstanz gelöscht wird;
- Name-Change-Meldung: benachrichtigt bei der Namensänderung eines verwalteten Objekts alle relevanten Management-Instanzen;
- Attribute-Value-Change-Meldung: informiert über die Modifikation von Attributwerten in einem verwalteten Objekt.

| SMI | Pass-Through | CMIS |
|---|---|---|
| Create | PT-CREATE | M-CREATE |
| Delete | PT-DELETE | M-DELETE |
| Action | PT-ACTION | M-ACTION |
| Replace | PT-SET | M-SET |
| Add | PT-SET | M-SET |
| Remove | PT-SET | M-SET |
| Replace-with-Default | PT-SET | M-SET |
| Get | PT-GET | M-GET |
| Notification | PT-EVENT | M-EVENT-REPORT |

**Tab. 3. Pass-Through-Dienste:** Dargestellt sind die Pass-Through-Dienste, die die allgemeinen SMI-Operationen auf die zugrundeliegenden CMIS-Operationen oder CMIS-Dienste abbilden.

Die beiden im Rahmen dieser Funktion in [ISO10164-1] definierten Funktionseinheiten sind *All Notifications* und *Pass Through* [SITJ91].

#### 2.5.6.2 Zustand-Management-Funktion

Die Zustand-Management-Funktion ist in [ISO10164-2] definiert worden und stellt ein allgemeines Modell zur Beschreibung von Zuständen der Ressourcen (verwalteten Objekten) dar. Unter dem Zustand einer Ressource wird allgemein die Verfügbarkeit oder Betriebsbereitschaft verstanden. Das Modell spezifiziert die Zustände einer Ressource grob durch die drei Attribute

- Betriebszustand *(operational state)*,
- Nutzungszustand *(usage state)* und
- Verwaltungszustand *(administrative state)*.

Außerdem werden innerhalb dieser Funktion weitere zusätzliche Attribute definiert:

- Status-Attribute *(status attributes)*,
- Management-Zustandsattribut *(management state attribute)* und
- Zustand-Attribut-Gruppe *(state attribute group)*.

Das ***Betriebszustandsattribut*** wird zur Darstellung der Betriebsbereitschaft einer Ressource verwendet. Dabei kann das Attribut die zwei folgenden Werte gemäß Abb. 15 besitzen.

- ENABLED: zeigt die teilweise oder vollständige Verfügbarkeit einer Ressource an.
- DISABLED: beschreibt die vollständige Nichtverfügbarkeit einer Ressource [Rose90b].

Das *Nutzungszustandsattribut* dient dazu, die aktuelle Nutzung einer Ressource zu beschreiben. Erlaubt eine Ressource einen Mehrbenutzer-Modus, so darf dann von einem Grad der Nutzung gesprochen werden. Gemäß Abb. 16 kann dieses Attribut die vier folgenden Werte annehmen:

- IDLE: tritt auf, wenn die Ressource nicht benutzt wird;
- BUSY: wird benutzt, um die Beschäftigung einer Ressource im Einbenutzer-Modus oder die volle Nutzung der Kapazität einer Ressource im Mehrbenutzer-Modus anzuzeigen;
- ACTIVE: zeigt an, daß eine Ressource im Mehrbenutzer-Modus zur Zeit benutzt wird und noch ungenutzte Kapazität besitzt;
- UNKNOWN: liegt vor, wenn keine Informationen über den aktuellen Nutzungsgrad der Ressource vorliegen [Rose90b].

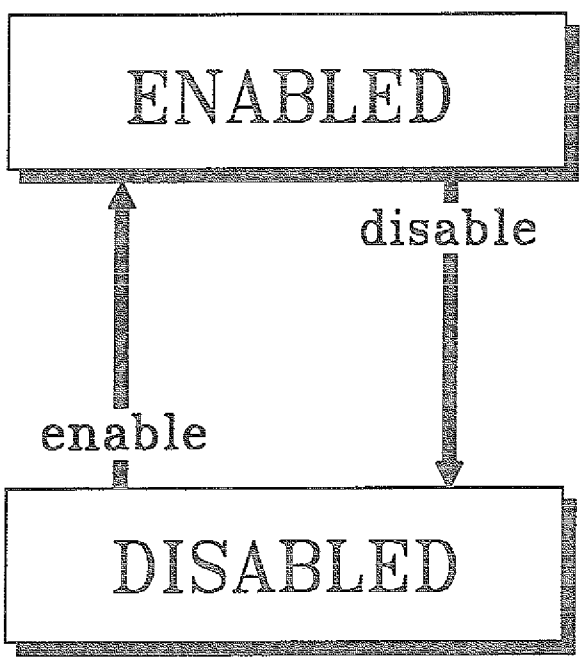



Abb. 15. Betriebszustände: Dargestellt sind die Betriebszustand-Attributwerte und die möglichen Übergänge.

Das *Verwaltungsattribut* dient der Darstellung der durch die Verwaltungsaktivitäten auftretenden Zustände. Unter Verwaltungsaktivitäten versteht man z.B. die gezielte Sperrung einer Ressource zur Benutzung im Rahmen des Managements von offenen Systemen. Das Verwaltungszustand-Attributwert wird, im Gegensatz zum Nutzungszustand- oder Betriebszustand-Attributwert einer Ressource, durch den Administrator gesetzt und ist sonst unabhängig von der Nutzung und der Betriebsbereitschaft der Ressource. Während der Betriebs- und Nutzungs-Zustand einer Ressource nur gelesen werden dürfen, da der Betriebszustand eines Elements und das Hinzukommen neuer Benutzer nicht vom Administrator beeinflußt werden, ist beim Verwaltungszustandsattribut sowohl das Lesen als auch das Schreiben erlaubt. Dieses Attribut besitzt drei Werte, die in der Abb. 17 dargestellt sind:

- UNLOCKED: liegt für eine Ressource vor, die für die Verwaltungsaktivitäten nicht gesperrt ist;

- SHUTTING DOWN: kennzeichnet eine Ressource, die zum Zweck der Verwaltungsaktivitäten gesperrt worden ist. Die Ressource bedient aber die Benutzer, die zur Zeit der Sperrung ihre Dienste in Anspruch genommen haben;
- LOCKED: bezeichnet eine Ressource, die zu Verwaltungsaktivitäten gesperrt worden ist und keinen Benutzern mehr zur Verfügung steht.

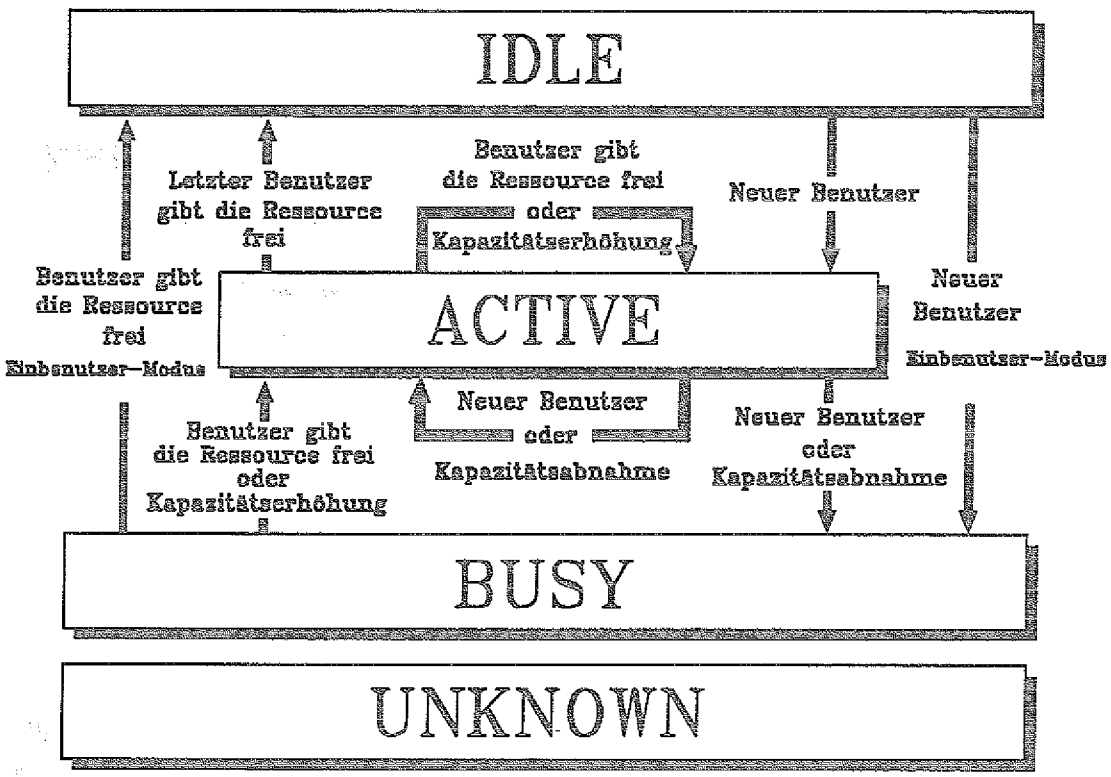



Abb. 16. Nutzungszustände: Dargestellt sind die Nutzungszustand-Attributwerte und die möglichen Übergänge.

Die Zustand-Management-Funktion definiert auch die erlaubten Kombinationen der Zustände entsprechend der Tab. 4 und die erlaubten Übergänge zwischen den Zuständen.

| Mögliche Kombinationen | UNLOCKED | SHUTTING DOWN | LOCKED |
|---|---|---|---|
| DISABLED | IDLE | - | IDLE |
| ENABLED | IDLE | ACTIVE | IDLE |
| | ACTIVE | | |
| | BUSY | BUSY | |
| | UNKNOWN | | |

Tab. 4. Erlaubte Zustände für ein verwaltetes Objekt: Die möglichen Nutzungszustand-Attributwerte für ein verwaltetes Objekt werden in der Tabelle durch die Betriebszustand- und Verwaltungszustand-Attributwerte festgelegt. Außerdem stellt die Tripelkombination von Attributwerten einen Attributwert für das Management-Zustands-Attribut dar.

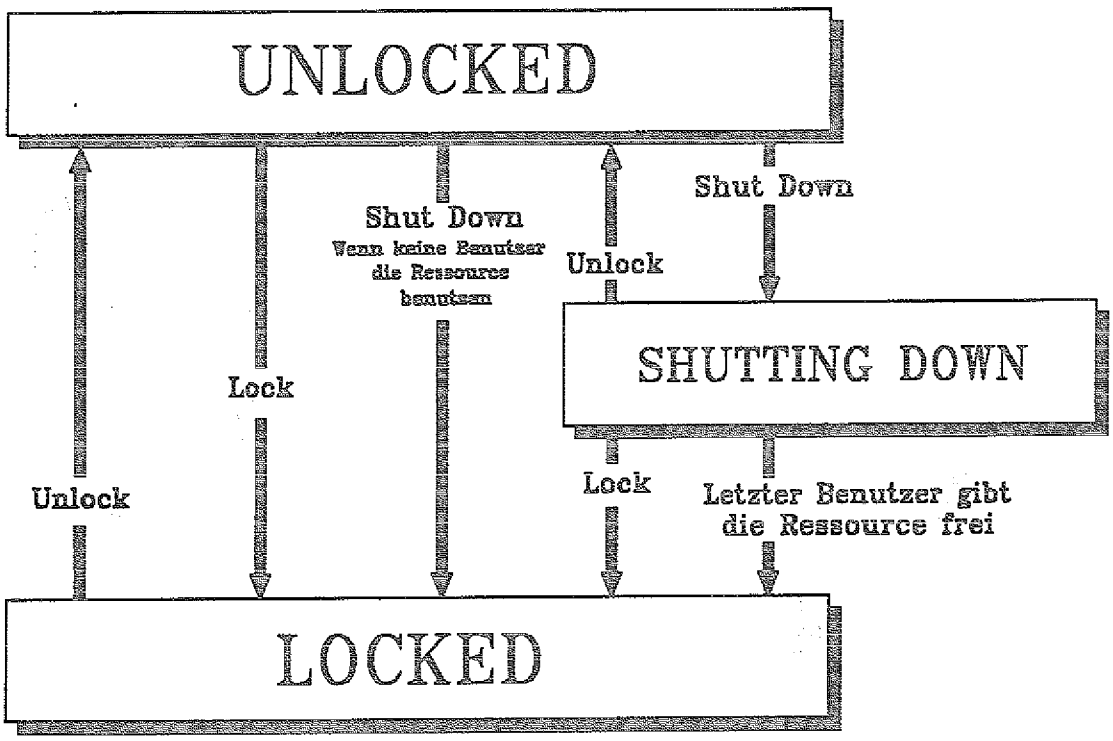



**Abb. 17. Verwaltungszustände:** Beschrieben sind die Verwaltungszustand-Attributwerte und die möglichen Übergänge.

Zu den Definitionen innerhalb der Zustand-Management-Funktion gehört auch die Definition der folgenden *Status-Attribute* und deren mögliche Werte, die eine genauere Beschreibung des aktuellen Zustands einer Ressource in standardisierter Form ermöglichen:

- REPAIR STATUS: gibt an, daß eine Ressource als in Reparatur anzusehen ist. Die beiden Attributwerte sind:
  - UNDER REPAIR: Ressource ist zur Zeit in Reparatur;
  - FAULT REPORT OUTSTANDING: Ein Fehler ist entdeckt und gemeldet worden, aber die Bestätigung der Meldung ist noch nicht zurückgekommen.
- INSTALLATION STATUS: kennzeichnet die Installation einer Ressource. Es existieren die folgenden Attributwerte:
  - NOT INSTALLED: Die Ressource ist noch nicht oder nicht vollständig installiert worden;
  - INITIALISATION INCOMPLETE: Die Ressource ist bereits installiert und mit ihrer Initialisierung wurde begonnen. Sie ist jedoch noch nicht abgeschlossen;

- INITIALISATION REQUIRED: Die Initialisation der Ressource ist noch nicht erfolgt.

- AVAILABILITY STATUS: Durch die Werte dieses Attributes werden zusätzliche Informationen über die Verfügbarkeit einer Ressource vermittelt. Die möglichen Werte des Attributes sind:

  - FAILED: Ressource ist wegen eines Fehlers nicht erreichbar;

  - IN TEST: Ressource ist im Test;

  - POWER OFF: Ressource ist ausgeschaltet;

  - OFF LINE: Ressource muß vor dem Betrieb aktiviert werden;

  - OFF DUTY: Ressource ist zeitweise nicht verfügbar;

  - DEPENDENCY: Ressource ist zum Betrieb von einer anderen zur Zeit nicht verfügbaren Ressource abhängig;

  - DEGRADED: Ressource kann ihren Dienst nicht mehr voll zur Verfügung stellen.

- CONTROL STATUS: liefert zusätzliche Angaben über die Steuerung einer Ressource und kann folgende Werte annehmen:

  - SUBJECT TO TEST: kennzeichnet eine Ressource, die getestet werden soll;

  - READ-ONLY: weist auf die administrativ verbotene Schreib-Operation auf eine Ressource hin, wie z.B. Plattenspeicher;

  - PART OF SERVICES LOCKED: bezeichnet, daß bestimmte Dienste einer Ressource vom Manager gezielt gesperrt worden sind;

  - RESERVED FOR TEST: kennzeichnet eine Ressource, die zeitweise zum Zweck des Tests nicht verfügbar ist;

  - SUSPENDED: bezeichnet eine Ressource, die von einer Management-Instanz gesperrt worden ist.

Zusätzlich zu den oben genannten Attributen werden in [ISO10164-2] noch zwei weitere Attribute definiert:

- *Management-Zustands-Attribut*: Der Wert dieses Attributes setzt sich aus den Werten der Betriebszustands-, Nutzungszustands- und Verwaltungszustands-Attribute zusammen. Die möglichen Attributwerte können aus der Tab. 4 auf Seite 51 entnommen werden;

- *Zustand-Attribut-Gruppe*: Diese Attribut-Gruppe umfaßt das Management-Zustands-Attribut, jedes damit verbundenes Status-Attribut und jedes objektspezifisches Zustandsattribut.

### 2.5.6.3 Attribute für die Darstellung von Beziehungen

In einem offenen System stehen die verwalteten Objekte derart miteinander in Beziehung, daß eine Operation auf einem verwalteten Objekt die anderen verwalteten Objekte beeinflußt. Unter einer Beziehung wird hier ein Satz von Regeln verstanden, die die Wirkung einer Operation, die auf einem Teil eines offenen Systems ausgeübt wird, auf die anderen Teile des offenen Systems beschreiben. Die Funktion Attribute für die Darstellung von Beziehungen *(Attribute for Representing Relationships)* in [ISO10164-3] dient der Normierung solcher Beziehungen. Diese Funktion stellt unter anderem ein Modell dar, das die Beziehungen zwischen den verwalteten Objekten spezifiziert. Die Beziehungen zwischen den verwalteten Objekten werden allgemein folgendermaßen unterteilt:

- direkte und indirekte Beziehungen *(direct and indirect relationships)* und
- symmetrische und asymmetrische Beziehungen *(symmetric and asymmetric relationships)*.

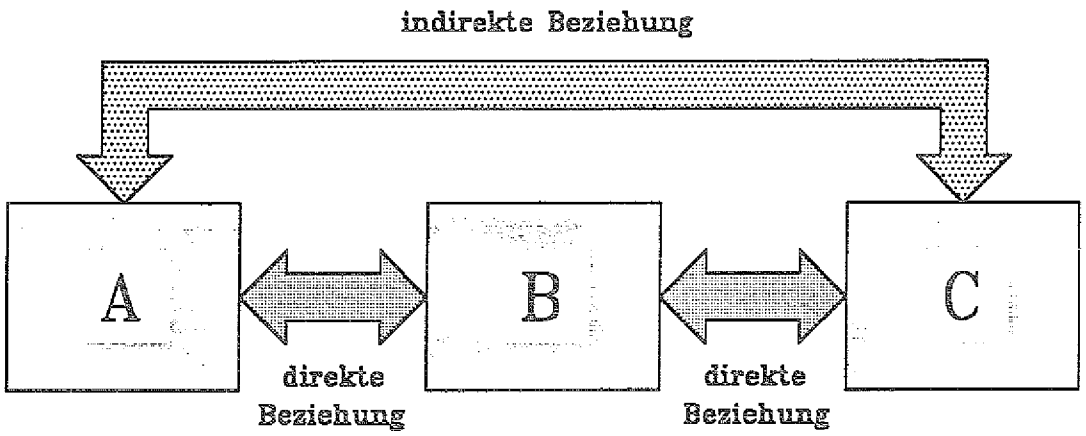



**Abb. 18. Direkte und indirekte Beziehung:** Das verwaltete Objekt A besitzt Informationen, die das verwaltete Objekt B spezifizieren und umgekehrt (direkte Beziehung). Daneben enthält das verwaltete Objekt B Informationen, die das verwaltete Objekt C kennzeichnen und umgekehrt (direkte Beziehung). Das verwaltete Objekt A steht mit dem verwalteten Objekt C indirekt über das verwaltete Objekt B in Beziehung. Ein verwaltetes Objekt, wie das verwaltete Objekt B, kann also auch zur Darstellung einer Beziehung benutzt werden, wenn es die notwendigen Informationen besitzt.

Außerdem werden im OSI-Netzwerkmanagement drei Kategorien von Beziehungen unterschieden:

- Enthaltungs-Beziehung *(containment relationship)*,
- Wechsel-Beziehung *(reciprocal relationship)* und
- Einseitig-Beziehung *(one-way relationship)*.

Eine Beziehung zwischen zwei verwalteten Objekten wird als ***direkte Beziehung*** bezeichnet, wenn ein Teil der Informationen, die einem verwalteten Objekt gehören, das andere

verwaltete Objekt ausdrücklich bezeichnen. Unter einer ***indirekten Beziehung*** versteht man dann eine Beziehung, die aus der Verkettung von zwei oder mehreren direkten Beziehungen entsteht (Abb. 18).

Die Beziehung zwischen zwei verwalteten Objekten z.B. X und Y ist ***symmetrisch***, wenn sie gleiche Rollen spielen und wenn die allgemeinen Sätze von Regeln, die die Beziehung X zu Y und Y zu X festlegen, gleich sind. Bei einer ***asymmetrischen Beziehung*** sind die allgemeinen Sätze von Regeln und die Rollen unterschiedlich.

Die ***Enthaltungs-Beziehung*** ist in [ISO10165-1] definiert und in dieser Arbeit in „2.4.2.3 Verwaltetes Objekt" auf Seite 17 beschrieben worden.

Die ***Wechsel-Beziehung*** stellt eine Bindung zwischen zwei verwalteten Objekten derart dar, daß jedes verwaltete Objekt den Namen des anderen als Beziehungsattributwert *(relationship attribute value)* enthält. Sind mehrere Namen als Wert des Beziehungsattributes *(set-valued attribute)* eingetragen, so existiert die Wechsel-Beziehung zwischen diesem Objekt und den anderen durch ihre Namen bezeichneten Objekten (Abb. 19).

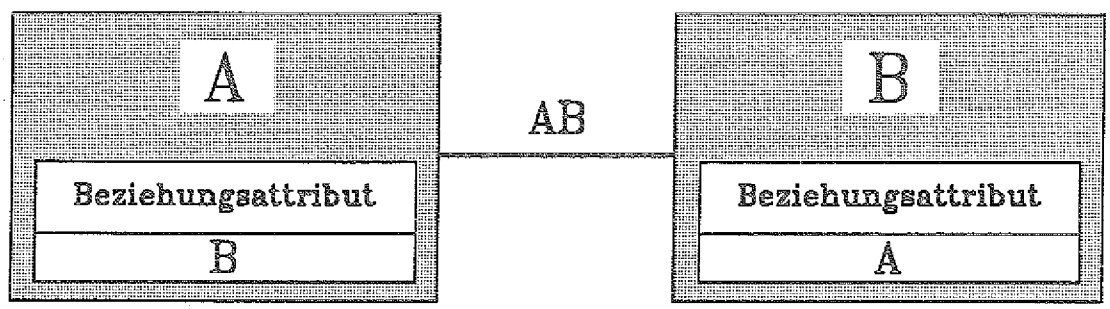



**Abb. 19. Wechsel-Beziehung:** Die verwalteten Objekte A und B stehen in einer direkten Beziehung zueinander, da A als Beziehungsattributwert den Namen B, und B als Beziehungsattributwert den Namen A besitzt. Die Wechsel-Beziehung wird durch die Namen der in Beziehung stehenden verwalteten Objekte und die beiden Beziehungsattributwerte, die die Beziehung darstellen, bezeichnet.

Die verschiedene Typen von Wechsel-Beziehungen werden im folgenden kurz beschrieben:

- Dienst-Beziehung *(service relationship)*: besteht zwischen zwei verwalteten Objekten, wenn ein verwaltetes Objekt dem anderen einen oder mehrere Dienste zur Verfügung stellt. Sie gehört zu den asymmetrischen Beziehungen.

- Partner-Beziehung *(peer relationship)*: beschreibt die Regeln, nach denen zwei gleichartige verwaltete Objekte miteinander kommunizieren. Sie stellt eine symmetrische Beziehung dar.

- Bereitschaft-Beziehung *(fallback relationship)*: Diese asymmetrische Beziehung zwischen zwei verwalteten Objekten bezeichnet das zweite verwaltete Objekt *(secondary object)* als das nächste zu bearbeitende verwaltete Objekt, wenn das erste

verwaltete Objekt ausfällt. Kommen mehrere verwaltete Objekte in Frage, so werden sie mit Prioritäten gekennzeichnet.

- Sicherstellung-Beziehung *(backup relationship)*: Diese asymmetrische Beziehung besteht zwischen zwei verwalteten Objekten nur dann, wenn das zweite Objekt als Reserve-Objekt gerade dem ersten sichergestellten Objekt *(backed up Object)* Reserve-Dienst zur Verfügung stellt. Das erste verwaltete Objekt ist dann im DISABLED-Zustand.

- Verbund-Beziehung *(group relationship)*: Eine Beziehung zwischen zwei verwalteten Objekten wird Verbund-Beziehung genannt, wenn ein verwaltetes Objekt einer Gruppe angehört *(member object)*, die durch das andere verwaltete Objekt repräsentiert wird *(owner object)*.

Die ***Einseitig-Beziehung*** zwischen zwei verwalteten Objekten ist eine asymmetrische Beziehung. Sie wird durch den Wert des Beziehungsattributes, das für diese Beziehung in nur einem verwalteten Objekt vorhanden ist, spezifiziert. Sie wird durch den Typ der Beziehung und den Namen des zugehörigen verwalteten Objekts dargestellt.

Die verwalteten Objekte können selbst eine Beziehung darstellen. Das verwaltete Objekt B in Abb. 18 ist ein Beispiel eines solchen verwalteten Objekts. Es steht mit zwei anderen verwalteten Objekten A und C in direkter Beziehung und kann eine indirekte Beziehung zwischen den verwalteten Objekten A und C repräsentieren. Dazu muß das verwaltete Objekt B Informationen enthalten, die den Typ der Beziehung und die anderen Beziehungsattribute der enthaltenen Beziehungen angeben.

Diese Funktion definiert außer den Meldungen, die eine Beziehungsänderung übertragen, einen Beziehungsänderung-Meldungsdienst *Relationship Change Reporting Service*, der die Änderungen in Beziehungsattributwerten von verwalteten Objekten überträgt. Seine Parameter sind identisch mit denen des M-EVENT-REPORT-Dienstes. Der Beziehungsänderung-Meldungsdienst kann im Bestätigt- oder Unbestätigt-Modus aufgefordert werden.

#### 2.5.6.4 Alarmmeldungs-Funktion

Die in einem offenen System aufgetretenen Fehler müssen der Management-Instanz gemeldet werden. Die Form der Fehlermeldungen ist in einem heterogenen Netz in der Regel vom verwalteten Objekt abhängig. Die Aufgabe der Alarmmeldungs-Funktion [ISO10164-4] besteht in der Bereitstellung standardisierter Fehlermeldungen, d.h. eines Satzes von Meldungstypen mit normierten Parametern. Für diesen Zweck stellt diese Funktion ein Modell zur Verfügung, das der Erkennung und Meldung von Fehlern dient. Durch dieses Modell werden die fünf folgenden Alarmtypen spezifiziert:

1. Kommunikations-Alarmtyp *(communications alarm type)*: ist für die Probleme vorgesehen, die bei Prozessen und Prozeduren der Datenübertragung auftreten.

2. Dienstgüte-Alarmtyp *(quality of service alarm type)*: zeigt die Abnahme der Qualität eines Dienstes an.

3. Verarbeitungs-Alarmtyp *(processing alarm type)*: kennzeichnet die Fehler, die bei der Software oder Datenverarbeitung auftreten.

4. Geräte-Alarmtyp *(equipment alarm type)*: steht für die Meldung eines Gerätefehlers zur Verfügung.

5. Umgebungs-Alarmtyp *(environmental alarm type)*: zeigt alle Probleme an, die in Verbindung mit der Umgebung eines Gerätes auftreten, wie z.B. Ausfall der Klimaanlage.

Außerdem wird durch dieses Modell allgemein die Auswirkung eines Fehlers auf die dem Benutzer zur Verfügung stehende Dienstqualität in vier Stufen, CRITICAL, MAJOR, MINOR und WARNING, kategorisiert. Diese Stufen werden weiter unten innerhalb des Dringlichkeitsgrad-Parameters definiert. Die fünf oben genannten allgemeinen Alarmtypen enthalten die folgenden Parameter:

- Wahrscheinliche Ursache *(probable cause)*: spezifiziert die Alarmursachen. Eine Liste der Alarmursachen ist in [ISO10164-4] angegeben, die durch Addenda zu diesem Standard ergänzt werden dürfen. Außerhalb dieses Standards dürfen weitere mögliche Alarmursachen nur dann definiert werden, wenn sie den Registrationsprozeduren folgen, die für ASN.1-Objektbezeichner-Werte definiert sind. Als Beispiele können Prozessor-Probleme, Signalverlust und Software-Fehler genannt werden.

- Spezifische Probleme *(Specific problems)*: Die wahrscheinlichen Fehlerursachen können durch diesen optionalen Parameter zusätzlich spezifiziert werden.

- Dringlichkeitsgrad: Fehler beeinflussen die Funktionalität von verwalteten Objekten. Um den Grad dieser Beeinflussung und dadurch die Dringlichkeit der Alarme festzulegen, wird durch diesen Parameter die Heftigkeit der Auswirkung in sechs Stufen unterteilt. Die vier Stufen CRITICAL, MAJOR, MINOR und WARNING beschreiben den Beeinträchtigungsgrad des einem Benutzer bereitgestellten Dienstes. Die sechs Stufen werden wie folgt beschrieben:

  1. CLEARED: weist auf das Löschen von einem oder mehreren gemeldeten Alarmen hin;

  2. INDETERMINATE: zeigt an, daß der Beeinträchtigungsgrad der Funktionalität eines verwalteten Objekts nicht bestimmt werden kann;

  3. CRITICAL: zeigt an, daß aufgrund der Existenz der dienstbeeinträchtigenden Zustände sofortige Maßnahmen dagegen notwendig sind. Ein verwaltetes Objekt, das ausgefallen ist und dessen Funktionalität wieder hergestellt werden muß, befindet sich z.B. auf dieser Stufe;

  4. MAJOR: macht kenntlich, daß sich ein dienstbeeinträchtigender Zustand entwickelt hat und Behebungsmaßnahmen ergriffen werden müssen. Diese Stufe ist z.B. für ein verwaltetes Objekt dann erreicht, wenn sein Dienstangebot stark reduziert ist und wieder hergestellt werden muß;

5. MINOR: beschreibt einen Fehler, der den dem Benutzer zur Verfügung stehenden Dienst nicht beeinflußt. Der Fehler sollte aber zur Vermeidung von Folgefehlern, die möglicherweise den bereitgestellten Dienst beeinflussen können, beseitigt werden;

6. WARNING: kennzeichnet einen dienstbeeinträchtigenden Fehler, der noch keine bedeutende Auswirkung gehabt hat. Maßnahmen zur Diagnose und Behebung des Problems müssen ergriffen werden, um schwerwiegende Beeinträchtigungen des Dienstes zu vermeiden.

- Sichergestellter Zustand *(backed up status)*: Die Sicherstellung der Ersetzung eines ausgefallenen Objekts durch ein Reserve-Objekt wird durch diesen optionalen Parameter, der die Werte TRUE (sichergestellt) oder FALSE (nicht sichergestellt) besitzen kann, angezeigt.

- Reserve-Objektinstanz *(back up object instance)*: Ist der Parameter Sichergestellter Zustand vorhanden und besitzt er den Wert TRUE, so kann dieser optionale Parameter zur Kennzeichnung einer verwalteten Objektinstanz, die den Sicherstellungsdienst dem sicherzustellenden Objekt bereitstellt, verwendet werden. Dieser Parameter ist nützlich, wenn das Reserve-Objekt aus einer Grundmenge von Objekten genommen wird, die als Reserve-Objekte fungieren.

- Tendenzanzeige *(trend indication)*: zeigt die gegenwärtige Tendenz des Beeinträchtigungsgrads eines verwalteten Objekts an. Ist dieser Parameter vorhanden, so existieren ein oder mehrere unbearbeitete Alarme für das verwaltete Objekt, dem der Alarm mit diesem Parameter zuzuordnen ist. Der Parameter Tendenzanzeige besitzt die drei folgenden Werte:

  1. MORE SEVERE: zeigt an, daß der Grad der Dienstbeeinflussung eines verwalteten Objekts, der durch den Wert des Dringlichkeitsgrad-Parameters beschrieben wird, im gegenwärtigen Alarm höher als dergleichen in unbearbeiteten Alarmen ist;

  2. NO CHANGE: liegt vor, wenn der Grad der Dienstbeeinflussung eines verwalteten Objekts im gegenwärtigen Alarm und in unbearbeiteten Alarmen gleich sind;

  3. LESS SEVERE: ist vorhanden, wenn der Grad der Dienstbeeinflussung eines verwalteten Objekts im gegenwärtigen Alarm niedriger als dergleichen in unbearbeiteten Alarmen ist.

- Schwellinformation *(threshold information)*: Ist der Alarm infolge der Überschreitung eines Schwellenwertes entstanden, so wird dieser optionale Parameter, der aus den folgenden vier Unterparametern besteht, benötigt:

  1. Überschrittene Schwelle *(triggered threshold)*: ist der Bezeichner eines Schwellenattributes, das die Meldung verursacht hat;

  2. Schwellstufe *(threshold level)*: Liegt ein Meßelement *(Gauge)* vor, so werden durch Schwellstufe zwei Werte spezifiziert. Der erste bezeichnet den

überschrittenen Schwellenwert und der zweite den entsprechenden Hysteresewert. Im Falle eines Zählers wird durch die Schwellstufe nur ein Wert spezifiziert [ISO10164-4];

3. Beobachteter Wert *(observed value)*: bezeichnet den Wert des Meßelements oder Zählers, während die Schwelle überschritten wird und ist nicht dem Schwellenwert gleich, wenn das Meßelement diskrete Werte aufnehmen kann;

4. Aktivierungszeit *(arm time)*: bezeichnet für ein Meßelement den Zeitpunkt der Überschreitung des Hysteresewertes, während sie für einen Zähler den Zeitpunkt der letzten Initialisierung oder Ausgleichung angibt.

- Meldungsbezeichner *(notification identifier)*: Durch diesen optionalen Parameter wird ein Bezeichner für den Alarm bereitgestellt.

- Verkettete Meldung *(correlated notification)*: Dieser optionale Parameter enthält einen Satz von Meldungsbezeichnern und, wenn nötig, die jeweiligen Namen der verwalteten Objektinstanzen. Dieser Satz von Meldungsbezeichnern stellt die Menge aller Meldungen dar, die mit diesem Alarm in Beziehung stehen.

- Zustandsänderung *(generic state change)*: Ein mit einem Alarm verbundener Zustandsübergang wird durch diesen optionalen Parameter angezeigt, der aus zwei Unterparametern besteht:

  1. Alter Zustand *(generic old state)*: gibt den Zustand eines Objekts im Zeitpunkt des Auftretens eines Alarms an, und

  2. Neuer Zustand *(generic new state)*: bezeichnet gegenwärtigen Zustand eines Objekts.

- Überwachte Attribute *(monitored attributes)*: Dieser optionale Parameter dient der Identifizierung von einem oder mehreren Attributen eines verwalteten Objekts und ihrer entsprechenden Werte zur Zeit des Alarms. Hierdurch wird rechtzeitiges Berichten über vorherrschende Bedingungen zur Zeit des Alarms ermöglicht.

- Vorgeschlagene Reparaturaktionen *(proposed repair actions)*: Dieser optionale Parameter wird dann verwendet, wenn der Fehler bekannt ist, und das verwaltete System eine oder mehrere Lösungen vorschlagen kann.

- Problemtext *(problem text)*: Eine Beschreibung des zu meldenden Problems im Klartext wird durch diesen optionalen Parameter ermöglicht.

- Problemdaten *(problem data)*: Ein Satz von zusätzlichen Informationen, wie Diagnoseinformationen und andere den Alarm betreffenden Informationen, kann in diesem optionalen Parameter übertragen werden. Er ist nur für die Informationen vorgesehen, die nicht auf den anderen Parametern abgebildet werden können.

# 3 SNMP-Netzwerkmanagement

Das SNMP-Netzwerkmanagement, das vereinfacht nach seinem Kommunikationsprotokoll häufig SNMP *(Simple Network Management Protocol)* genannt wird, stellt den Internet-Standard für das Netzwerkmanagement der auf TCP/IP-basierenden Netzwerke dar [Cerf88, Cerf89]. Das SNMP-Netzwerkmanagement hat sich wegen seiner Einfachheit bezüglich der Architektur sehr schnell verbreitet und ist in vielen Umgebungen implementiert worden. So hat das SNMP-Netzwerkmanagement eine breite Akzeptanz gefunden und sich über die Grenzen der auf TCP/IP-basierenden Netzwerke ausgebreitet [Scot90a].
Das SNMP-Netzwerkmanagement wurde wie das OSI-Netzwerkmanagement für das Management heterogener Netzwerke entwickelt. Der Unterschied liegt aber im wesentlichen darin, daß das SNMP-Netzwerkmanagement die derzeitigen Forderungen der Administratoren erfüllt und auf bereits weit verbreiteten existierenden Netzwerkprotokollen basiert.

In diesem Kapitel werden daher analog zum zweiten Kapitel zuerst die die Netzwerkmanagement-Technologie normierende Instanz und der Standardisierungsprozeß beschrieben. Danach wird ein kurzer Überblick über die dem Netzwerkmanagement unterliegenden Protokolle gegeben, die für das Verständnis des hier untersuchten SNMP-Netzwerkmanagements von Bedeutung sind.

## 3.1 Internet-Standardisierung

Internet, auch DARPA-Internet oder TCP/IP-Internet genannt [Come88], ist das größte Netz von Netzwerken *(internet)* in der Welt; es umfaßt große nationale Netzwerke, wie z.B. MILNET *(MILitary NETwork)*, NSFNET *(National Science Foundation NETwork)* und CREN *(Corporation for Research and Educational Networking)*, sowie viele regionale und lokale Netzwerke. Internet basiert auf einem Protokollsatz, der weite Verbreitung gefunden hat und von vielen Herstellern unterstützt wird. Zur Bezeichnung aller Gruppen, die weltweit diesen Protokollsatz benutzen, wird der Begriff Internet-Gemeinschaft *(Internet Community)* verwendet. Der Internet-Protokollsatz wurde entwickelt, um auf unzuverlässigen Medien eine zuverlässige Übertragung der Informationen zu ermöglichen und der Interoperabilität zwischen den Netzwerken zu dienen. Dabei versteht man unter unzuverlässigen Medien solche, die Technologien, wie z.B. Ethernet, benutzen, die also keine Garantie für den Empfang der Informationen beim Empfänger sicherstellen [Rose90a].

Die Entwicklung von Internet-Protokollen wird durch die Instanz IAB *(Internet Activities Board)*, die sich hauptsächlich aus Internet-Entwicklern und -Implementatoren zusammensetzt, beaufsichtigt. IAB ist im Gegensatz zu ISO keine internationale Standardisierungsorganisation und produziert in der Regel wenige Dokumente selbst. IAB hat im wesentlichen die Aufgabe, die Entwicklung der Internet-Protokolle zu koordinieren, um dadurch die Interoperabilität zwischen den Protokollen zu erhöhen.
Innerhalb IAB existieren zwei Gruppen, IETF *(Internet Engineering Task Force)* und IRTF *(Internet Research Task Force)*, die für kurzfristige technische Aufgaben bzw. langfristige Forschung zuständig sind. IETF und IRTF werden jeweils durch die Steue-

rungsgruppen IESG *(Internet Engineering Steering Group)* und IRSG *(Internet Research Steering Group)* verwaltet.

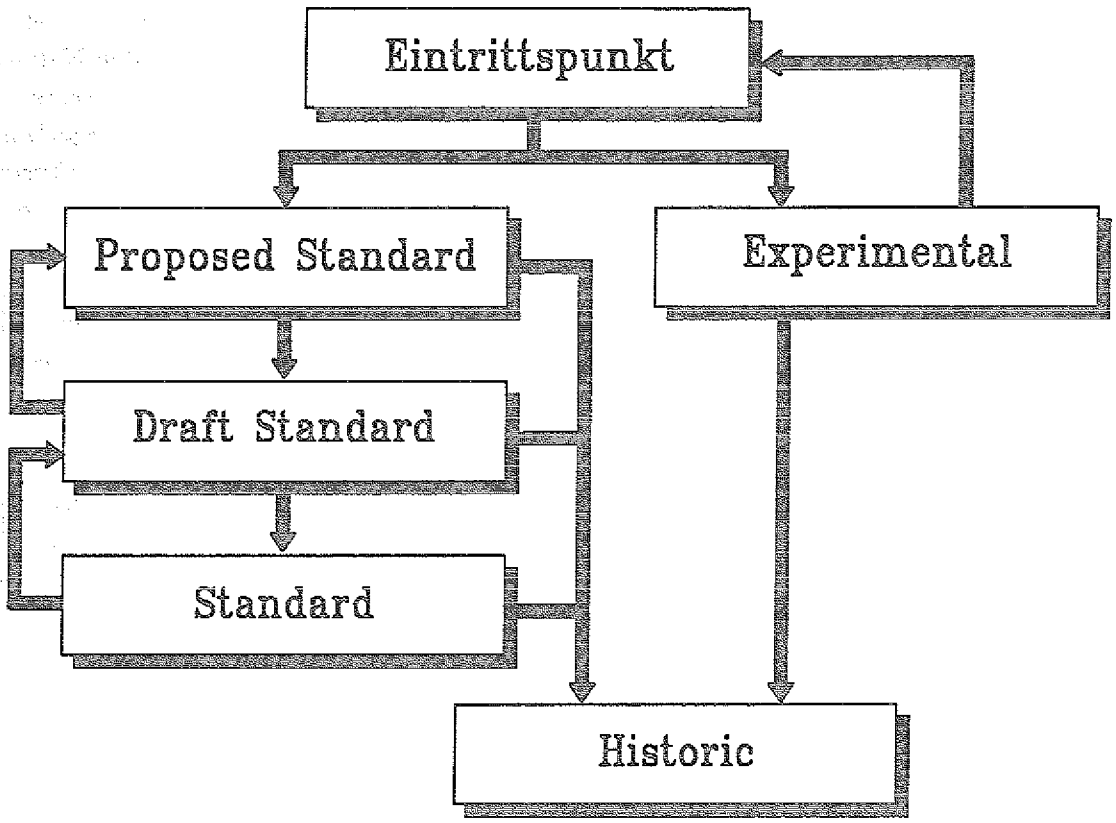



**Abb. 20. Standardisierungspfad:** Damit ein Dokument in den Standardisierungspfad eingeführt werden kann oder von einer Stufe zu einer anderen Stufe übergehen kann, sind die IESG-Empfehlung und die IAB-Ratifizierung erforderlich. Ist ein Dokument zur Standardisierung nicht bereit, so wird es der Stufe *Experimental* zugeordnet. Außerdem kann ein Dokument aus verschiedenen Gründen, wie z.B. Ersetzung durch ein anderes Dokument, in die Stufe *Historic* übergehen.

Die Entwicklungs- und Standardisierungsarbeit wird hauptsächlich in den Arbeitsgruppen von IETF durchgeführt. Außerdem können die Mitglieder der Internet-Gemeinschaft Protokolle für Internet entwerfen, dokumentieren, implementieren und testen. IAB stellt dabei die Forderung, daß die Protokolle in Form von RFCs *(Request For Comments)* dokumentiert werden müssen, unabhängig davon, ob die entstandenen Dokumente letztlich zur Standardisierung vorgeschlagen werden oder nicht.
Die Internet-Dokumente setzen sich hauptsächlich aus den beiden folgenden Dokumentenreihen zusammen:

1. *Internet Draft* und
2. *Request For Comment*.

*Internet Draft* ist ein Dokument, das keinen Standardisierungszustand besitzt und durch IETF-Arbeitsgruppe herausgegeben wird. Ein *Internet Draft* wird also als ein Bericht über eine laufende Arbeit betrachtet, worauf hier nicht genauer eingegangen werden soll.

Ein RFC ist ein Dokument, das jedes Thema der Rechnerkommunikation vom Sitzungsbericht bis zur Spezifizierung von Standards umfassen darf. Das bedeutet, daß „alle Standards als RFCs veröffentlicht werden, aber nicht alle RFCs Standards spezifizieren"[10].
Die RFCs sind in der Regel die Arbeitsnotizen der Internet-Forschungs- und Internet-Entwicklungs-Gemeinschaft *(Network Working Group)*. Außerdem kann jedes Mitglied der Internet-Gemeinschaft ein Dokument zur Veröffentlichung als RFC dem RFC-Herausgeber *(RFC Editor)* vorlegen. Durch RFCs werden also die Ideen einzelner Mitglieder in einer einfachen Weise allen Mitgliedern der Internet-Gemeinschaft zur Verfügung gestellt. Jedes RFC besitzt eine Nummer, die sich ändert, wenn wegen Verbesserungen oder Textkorrekturen eine neue Version herausgegeben wird. Die Nummern werden durch den RFC-Herausgeber *(RFC Editor)*, der ein Mitglied von IAB ist, den RFCs zugeteilt. Um die Eindeutigkeit der den RFCs zugeordneten Nummer zu gewähren, werden Listen von zulässigen Nummern durch IANA *(Internet Assigned Numbers Authority)* verwaltet. Jede neue Version von RFC trägt zur Vermeidung der Verwechselungen zusätzlich zur neuen Nummer die alte Nummer. Diese Vorgehensweise unterscheidet sich von der ISO/IEC-Methode derart, daß eine im Anfangsstadium einem Dokument zugeordnete Nummer nicht mehr geändert werden darf. Außerdem sind nicht alle entworfenen RFCs, im Gegensatz zu ISO-Dokumenten, zur Standardisierung vorgesehen.

Jedes RFC wird durch die zwei folgenden unabhängigen Attribute gekennzeichnet:

1. Zustand *(state)* wird zur Bezeichnung der Standardisierungsstufe eines Dokumentes verwendet und kann folgende Werte besitzen (vgl. Abb. 20):

   - *Historic*: Das Dokument ist durch neue Entwicklungen ersetzt worden oder ist wegen Mangel an Interesse nicht mehr aktuell;
   - *Informational*: umfaßt alle Dokumente, die außerhalb des IAB-Wirkungsbereiches entwickelt worden sind und für die Internet-Gemeinschaft von Interesse sind;
   - *Experimental*: bezeichnet Dokumente, die als Teile eines laufenden Forschungsprojekts entwickelt werden und nur innerhalb des Versuchsbereiches implementiert werden dürfen;
   - *Proposed Standard*: kennzeichnet die Dokumente, die zur Standardisierung durch IAB vorgesehen sind, aber noch Verbesserungen nach Implementierungen und Prüfungen unterworfen sein können;
   - *Draft Standard*: Alle Dokumente, die die erste Standardisierungsstufe *Proposed Standard* nach IAB-Ratifizierung und IESG-Empfehlung zurückgelegt haben;
   - *Standard* sind die durch IAB gebilligten offiziellen Dokumente für Internet.

---

10 siehe Seite 4 in [IAB91]

2. Status *(status)*: beschreibt die Implementationsforderung eines Dokumentes und wird durch die folgenden Begriffe festgelegt:

   - *Required*: Dokument muß implementiert werden;
   - *Recommended*: Dokument soll implementiert werden;
   - *Elective*: Dokument darf implementiert werden;
   - *Limited Use*: Dokument ist nur begrenzt zu benutzen;
   - *Not Recommended*: Dokument wird zur allgemeinen Benutzung nicht empfohlen.

Die im Internet zur Standardisierung vorgesehenen Protokolle durchlaufen die drei Standardisierungsstufen:

1. *Proposed Standard*,
2. *Draft Standard* und
3. *Standard*.

Zum Übergang von einer Stufe zur anderen sind außer genauen Prüfungen und experimentellen Tests die Empfehlung von IESG und die Ratifizierung von IAB notwendig. Dabei wird eine Zeitspanne von vier Monaten für den Übergang von *Proposed Standard* zum *Draft Standard* und sechs Monaten für den Übergang von *Draft Standard* zum *Standard* eingehalten, damit die Internet-Gemeinschaft auf die Beschlüsse reagieren kann.

## 3.2 Internet-Protokollsatzmodell

Das Umfeld des SNMP-Netzwerkmanagements wird heute maßgeblich durch den Internet-Protokollsatz, auch DoD[11]- oder TCP/IP-Protokollfamilie genannt, bestimmt. In den letzten Jahren hat sich speziell für den Verbund heterogener Systeme die TCP/IP-Protokollfamilie des DoD durchgesetzt. Sie besteht aus einer Reihe von Protokollen für die Schichten 3-7 des OSI-Referenzmodells und wurde ursprünglich vom DoD zur Vereinheitlichung der Rechnerkommunikation im Rahmen des ARPANET[12] entwickelt [Kauf90b].

Analog zum OSI-Protokollsatz ist der Internet-Protokollsatz in Schichten organisiert. Es werden vier verschiedene Schichten mit den folgenden Bezeichnungen, die die Verbindung zum OSI-Referenzmodell aufzeigen, unterschieden [Rose91a] (vgl. Abb. 21):

---

11 Verteidigungsministerium der USA *(Department of Defense, DoD)*

12 ARPANET ist ein Paketvermittlungsnetz und ein Produkt von ARPA, jetzt DARPA genannt, *((Defense) Advanced Research Projects Agency)* des amerikanischen Verteidigungsministeriums.

- Schnittstellenschicht *(interface layer)*: entspricht den Schichten Bitübertragungs- und Sicherungsschicht des OSI-Referenzmodells und ist also für die Übertragung von Informationen auf der Hardware zuständig;
- Internetzschicht *(internet layer)*: entspricht der Vermittlungsschicht des OSI-Referenzmodells und stellt netzüberschreitende Vermittlungsdienste zur Verfügung. Sie realisiert daher das Konzept der netzüberschreitenden Kommunikation *(internetworking concept)*;
- Transportschicht *(transport layer)*: entspricht der Transportschicht des OSI-Referenzmodells und liefert Ende-zu-Ende-Dienste, die zur Bereitstellung eines zuverlässigen Transportmechanismus benutzt werden können;
- Anwendungsschicht *(application layer)*: entspricht den Schichten Sitzungs-, Darstellungs- und Anwendungsschicht des OSI-Referenzmodells und dient der Bereitstellung von Mechanismen, die den Endbenutzern Dienste liefern.

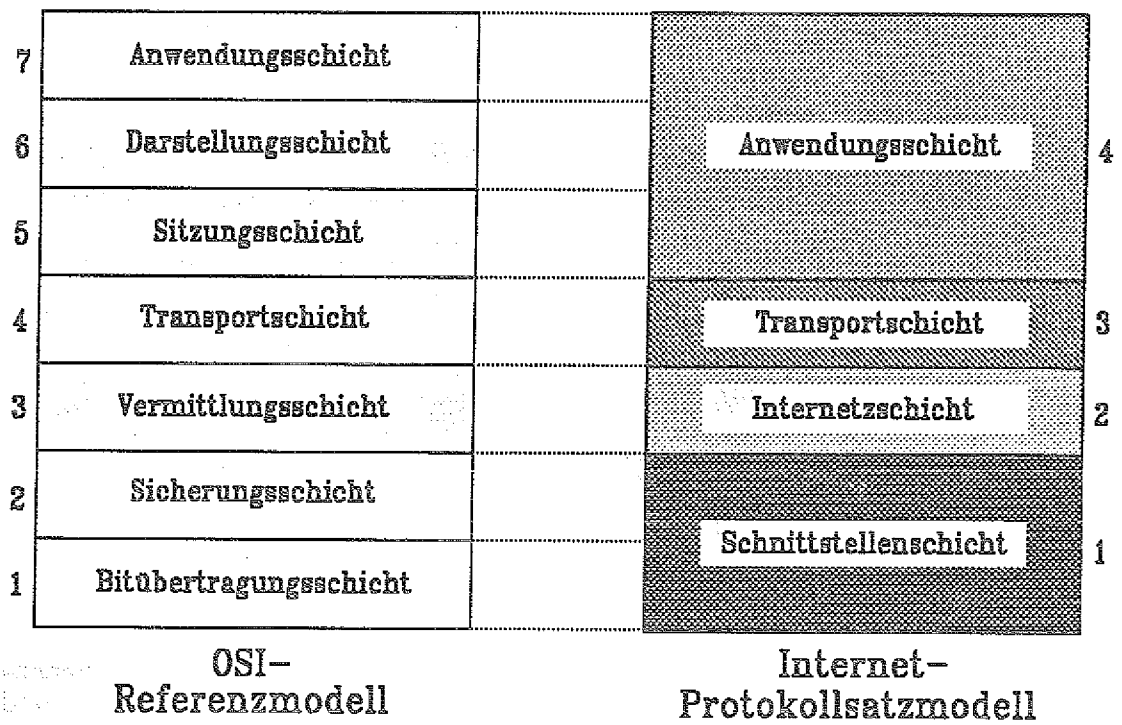



**Abb. 21. Internet-Protokollsatzmodell:** In Analogie zum OSI-Referenzmodell können den Internetprotokollen vier Schichten, d.h. Schnittstellen-, Internetz-, Transport- und Anwendungsschicht zugeordnet werden, die zusammen in dieser Arbeit das Internet-Protokollsatzmodell kennzeichnen.

Die Tatsache, daß der Internet-Protokollsatz das Konzept der netzüberschreitenden Kommunikation realisiert, zeichnet ihn besonders für eine heterogene Netzwerkumgebung aus. So ist diese Technologie auf verschiedenen Übertragungsmedien, -systemen und -netzen und auch auf verschiedenen Rechnern einsetzbar. Das Netzwerk besteht aus vielen verschiedenen Teilnetzen, die über Zwischensysteme, wie

z.B. Gateways, miteinander verbunden sind, und die Kommunikation unterschiedlicher Rechner verschiedener Hersteller miteinander erlaubt.

Die wichtigsten Internet-Protokolle auf den Schichten des Internet-Protokollsatzmodells werden im folgenden kurz beschrieben, um die Position des SNMP-Protokolls in diesem Modell besser verstehen zu können [Davi88, Sant90, Kauf90b] (vgl. Abb. 22):

*Schnittstellenschicht*

Auf der Schnittstellenschicht des Internet-Protokollsatzes werden keinerlei Festlegungen bezüglich der zu verwendenden Protokolle gemacht. Als Beispiele hierfür können Ethernet, X.25 und Token-Ring genannt werden.

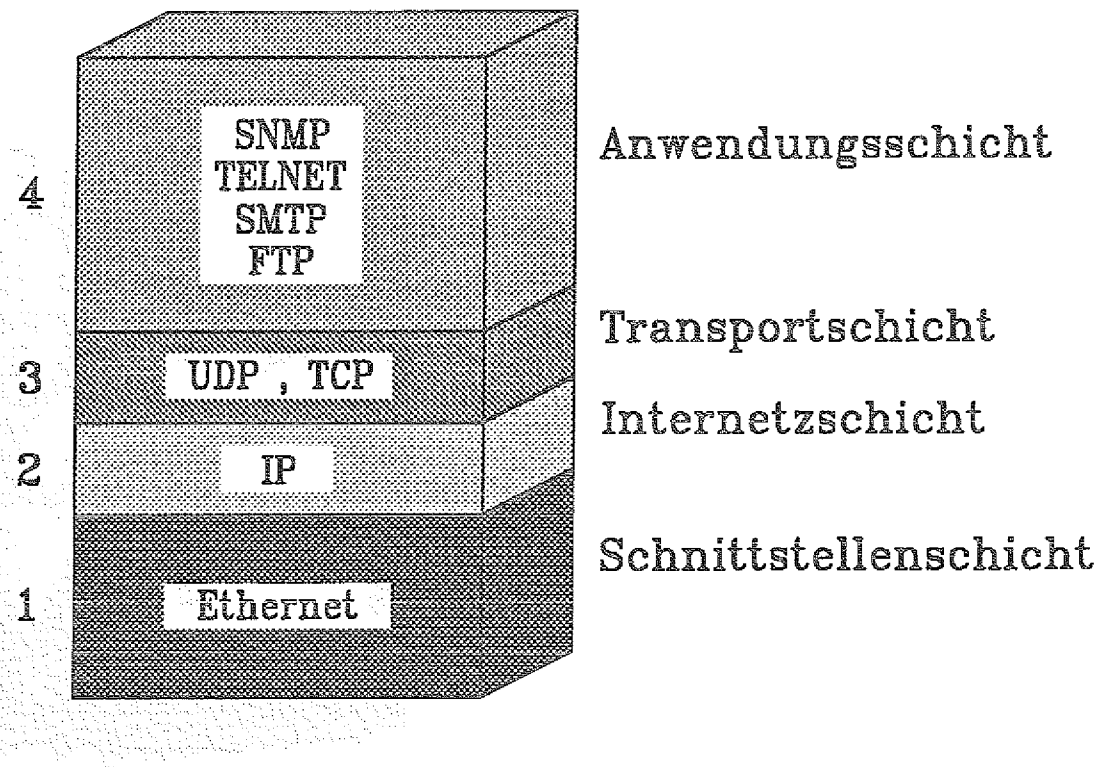



**Abb. 22. Wichtige Internet-Protokolle und -Anwendungen:** Dargestellt ist die Zuordnung der Internet-Protokolle und -Anwendungen zu den einzelnen Schichten des Internet-Protokollsatzmodells.

*Internetzschicht*

- Internet-Protokoll *(Internet Protocol, IP)*, beschrieben in *RFC791* [Post81a], ermöglicht den Austausch von Daten über mehrere Netze hinweg und bildet die Grundlage für die darüberliegenden Internet-Protokolle. IP realisiert die Funktionalität der Vermittlungsschicht des OSI-Referenzmodells, wobei seine Hauptaufgaben in der Adressierung innerhalb Internet, in der Wegewahl der Datagramme durch das Netzwerk *(Routing)* und in der Zerlegung und Wiederzusammensetzung großer Pakete bestehen. IP garantiert weder eine Ablieferung der

Datagramme beim Empfänger noch eine Einhaltung der Reihenfolge der Datagramme.

*Transportschicht*

- Das Protokoll TCP *(Transmission Control Protocol)*, beschrieben in *RFC793* [Post81b], wird als sicheres Host-zu-Host-Protokoll zwischen Hosts in paketorientierten Rechnerkommunikations-Netzwerken und in lokalen Netzwerken benutzt. TCP ist ein verbindungsorientiertes Ende-zu-Ende-Protokoll der Transportschicht des OSI-Referenzmodells und verwendet unter anderem einen einfachen, zuverlässigen Datagramm-Dienst von der unteren Vermittlungsschicht. Unter zuverlässigem Datagramm-Dienst soll ein Datagramm-Dienst verstanden werden, der einen Quittungsmechanismus zusammen mit einer Wiederholung verlorengegangener Pakete verwendet, um die sichere Ankunft der Informationen zu gewährleisten. Die guten Sicherungsfunktionen und die aufwendigen Wegewahlverfahren sind die wesentlichen Merkmale des Protokolls TCP.

- Das Protokoll UDP *(User Datagram Protocol)*, beschrieben in *RFC768* [Post80], erlaubt Anwendungsprozessen, ohne Aufbau einer virtuellen Verbindung, Datagramme auszutauschen. UDP ist im Vergleich zum TCP ein äußerst einfaches verbindungsloses Transportprotokoll und verwendet einen quittungslosen Datagramm-Austauschmechanismus. Andererseits basieren sowohl UDP als auch TCP auf dem Protokoll IP.

*Anwendungsschicht*

- Das Protokoll TELNET, beschrieben in *RFC854* [Post83], erlaubt die bidirektionale Byte-orientierte Kommunikation im Dialog mit anderen Systemen und stellt das Protokoll für Virtuelles Terminal im Internet-Protokollsatz dar. Hierbei wird den Benutzern eines Hosts erlaubt, sich in einem entfernten Host einzuloggen und so behandelt zu werden, als ob sie lokal eingeloggt wären. Außer der Host-Host-Verbindung sind Host-Prozeß-Verbindung oder Prozeß-Prozeß-Verbindung möglich, wobei die Prozesse nicht auf dem gleichen Host laufen dürfen.

- Das Protokoll FTP *(File Transfer Protocol)*, beschrieben in *RFC959* [Post85], steuert den Austausch von Dateien zwischen unterschiedlichen Systemen, sowie die indirekte Benutzung von Ressourcen anderer Rechner. Dabei werden zwei TCP-Verbindungen aufgebaut, die der getrennten Übertragung von Daten und Befehlen dienen. Die Befehle ermöglichen beispielsweise, daß die Dateien zwischen den Systemen kopiert oder gelöscht werden.

- Das Protokoll SMTP *(Simple Mail Transfer Protocol)*, beschrieben in *RFC821* [Post82], ist das Protokoll für die elektronische Post im Internet. SMTP regelt den Austausch von Briefen *(mails)* zwischen den Hosts, die entweder demselben lokalen Netz angehören oder über einen Gateway miteinander verbunden sind.

## 3.3 SNMP-Netzwerkmanagement-Architektur

In dieser Arbeit stellt die SNMP-Netzwerkmanagement-Architektur den Oberbegriff für SNMP-Netzwerkmanagement-Modell und SNMP-Netzwerkmanagement-Rahmenwerk dar. Das SNMP-Netzwerkmanagement-Modell beschreibt dabei die im SNMP-Netzwerkmanagement-System existierenden wesentlichen Komponenten und ihre Wechselwirkungen. Die Mechanismen und Festlegungen, wie z.B. die Definition von Objekten innerhalb des Netzwerkmanagement-Systems oder der Protokollaufbau, werden mittels dreier normierter Dokumente definiert, die das SNMP-Netzwerkmanagement-Rahmenwerk spezifizieren. Das Rahmenwerk gibt also den dokumentarischen Aufbau des Netzwerkmanagements an. Das SNMP-Netzwerkmanagement-Modell und das SNMP-Netzwerkmanagement-Rahmenwerk werden weiter unten eingehend erläutert.

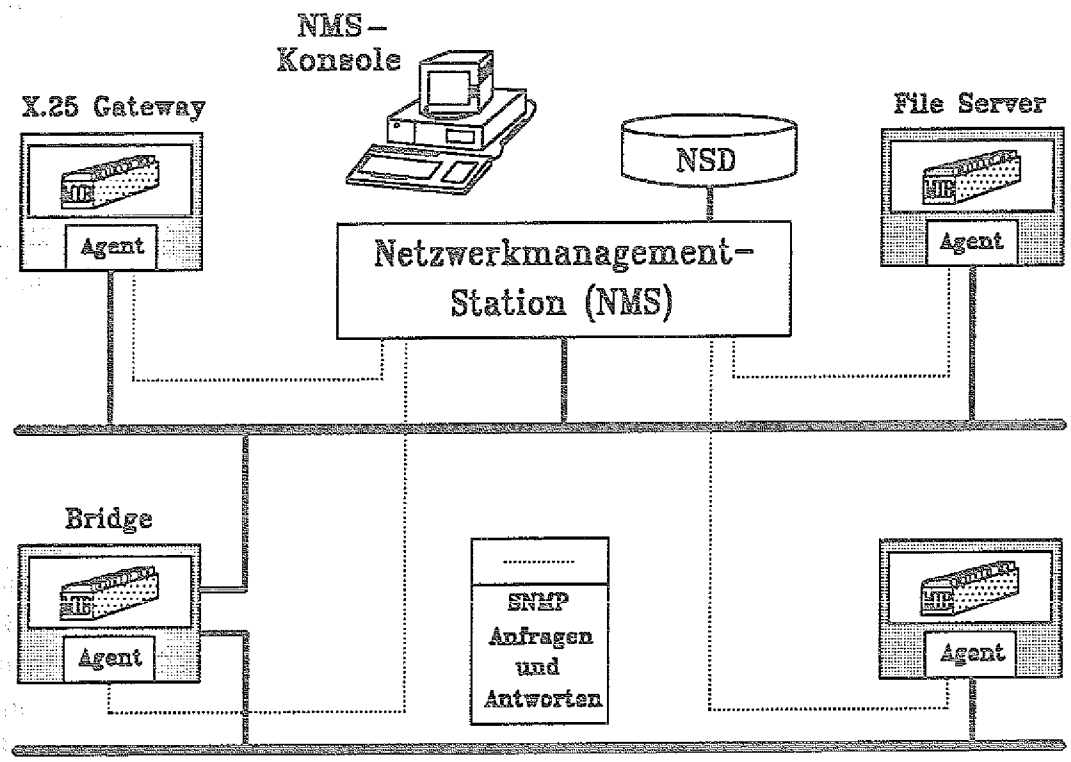



**Abb. 23. SNMP-Netzwerkmanagement-Modell:** Die NMS-Konsole ermöglicht den zentralen Überblick über das Netzwerk. Die Management-Informationen werden durch Agenten gesammelt und in den jeweiligen MIBs der Netzwerkgeräte bereitgehalten. Außerdem werden die Daten zentral in der NSD *(Network Statistics Database)* zu Korrelations- und Planungszwecken gespeichert. Die NMS kann dann die Management-Informationen lesen und ändern und dadurch das Netzwerk überwachen und steuern.

## 3.4 SNMP-Netzwerkmanagement-Modell

In einem System, das durch das SNMP-Netzwerkmanagement überwacht und gesteuert wird, können drei Komponenten unterschieden werden [Simp90, Mard91, CFSD89]:

1. Netzwerkmanagement-Station *(Network Management Station, NMS)*,
2. verwaltete Knoten *(managed nodes)*, die jeweils einen Agenten *(agents)* enthalten, und
3. das Netzwerkmanagement-Protokoll SNMP.

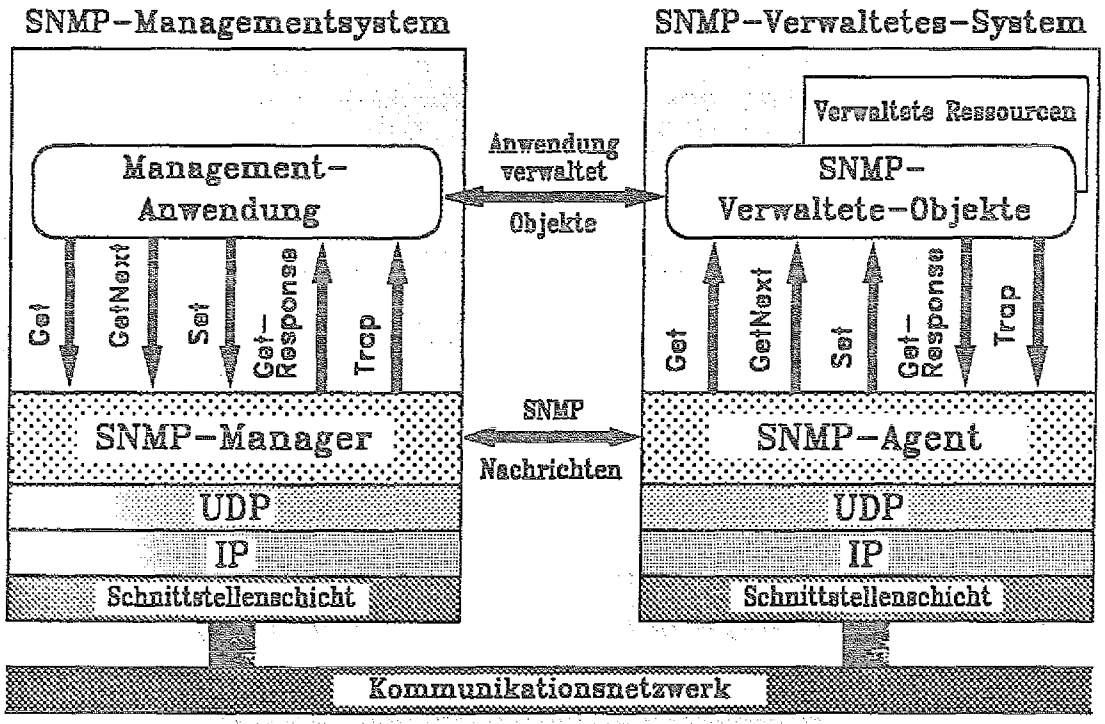



Abb. 24. **SNMP-Manager und -Agent:** Eine Management-Anwendung verwaltet die Netzwerkressourcen über den SNMP-Manager, der seinerseits durch den SNMP-Agenten das Netzwerk überwacht und steuert.

### 3.4.1 Netzwerkmanagement-Station

Unter einer Netzwerkmanagement-Station versteht man ein System, das für das Management eines Netzwerks oder Teile davon zuständig ist. In einem solchen System werden das Netzwerkmanagement-Protokoll SNMP und verschiedene Netzwerkmanagement-Anwendungen implementiert. Während das SNMP das Werkzeug für das Management bereitstellt, verwalten die Netzwerkmanagement-Anwendungen das Netzwerk, indem sie die Managementstrategie festlegen.

Der *Manager* ist z.B. ein in der Netzwerkmanagement-Station angesiedeltes Software-Programm und besitzt die Fähigkeit, den Agenten mittels verschiedener SNMP-Kommandos Fragen zu stellen, die der Netzwerkmanagement-Strategie entsprechen (vgl. Abb. 24) [Sabo91]. Das heißt, daß die Netzwerkmanagement-Stationen die Netzwerkmanagement-Anwendungen ausführen, um die Netzwerkelemente überwachen und steuern zu können.

Die Netzwerkmanagement-Station ist also eine zentrale Komponente, üblicherweise eine Workstation, die dem Administrator einen Überblick über den Zustand des Netzes verschafft und ihm Möglichkeiten zum Eingriff gibt.

### 3.4.2 Verwaltete Knoten

Ein verwalteter Knoten bezeichnet jedes im Netzwerk existierende zu verwaltende Gerät und setzt sich konzeptionell aus folgenden drei Komponenten zusammen [Rose91a, CaMa91] (vgl. Abb. 25):

- Arbeitsprotokolle *(useful protocols)*: dienen der Ausführung der eigentlichen Aufgabe des Geräts, wie z.B. die Protokolle IP und UDP;
- Management-Agent *(management agent)*: erlaubt die Überwachung und Steuerung von verwalteten Knoten, indem er auf die Anfragen der NMS unter Zuhilfenahme des Protokolls SNMP antwortet;
- Management-Instrumente *(management instrumentation)*: ermöglichen die Überwachbarkeit und Steuerbarkeit des Geräts, wie z.B. Zähler, Zustandsvariable und Tabellen.

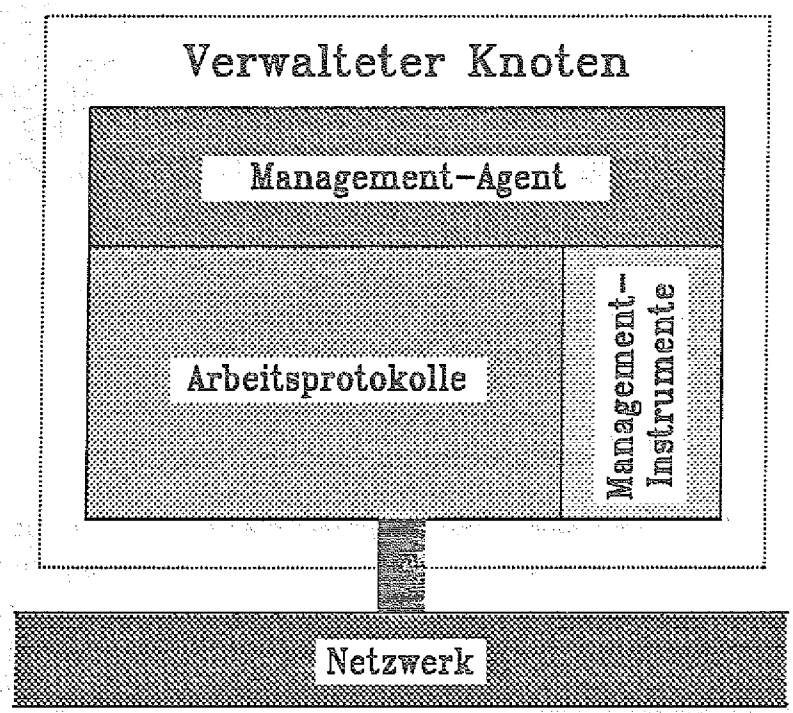



**Abb. 25. Verwalteter Knoten:** Dargestellt ist der konzeptionelle Aufbau eines verwalteten Knotens.

Der Management-Agent, auch kurz Agent genannt, ist ein Softwareprogramm. Jeder Agent besteht aus zwei Komponenten:

- Protokollmaschine *(protocol engine)*: implementiert das Netzwerkmanagement-Protokoll SNMP und kommuniziert mit einer SNMP-Anwendung in NMS mittels SNMP-Nachrichten[13];

- Managementprofil *(management profile)*: spezifiziert eine Teilmenge der Netzwerkmanagement-Objektinstanzen, die auch *MIB view* genannt wird, und implementiert Steuerungsmechanismen für berechtigten Zugriff[14].

Die Management-Daten, die für die Ausführung der Management-Funktionen wichtig sind, wie z.B. Zustandswerte, werden durch die Agenten mittels Management-Instrumenten gesammelt, vor Ort in einer MIB gespeichert und der NMS verfügbar gemacht. Unaufgeforderte Meldungen, im SNMP *Traps*[15] genannt, werden von den Agenten nur im Rahmen kritischer Bedingungen, wie z.B. außergewöhnlicher Fehler und Stromversorgungsausfällen erzeugt. So führt ein Agent die Anweisungen der Netzwerkmanagement-Station aus und erzeugt dadurch die Möglichkeit der Steuerung und Überwachung des Gerätes im Rahmen der Netzwerkmanagement-Strategie.
Verwaltete Knoten können Endsysteme, wie z.B. Hosts, Drucker, Terminalserver, Workstations, Zwischensysteme, wie z.B. Gateways und Routers oder das Kommunikationsnetzwerk unterstützende Geräte, wie z.B. Bridges und Multiplexer sein.

### 3.4.3 Netzwerkmanagement-Protokoll

Das Netzwerkmanagement-Protokoll SNMP basiert auf dem Konzept des Ferntestens *(remote debugging)*. Die verwalteten Knoten werden hierbei als eine Ansammlung von Objektinstanzen, die auch Variablen genannt werden, betrachtet. Ein verwalteter Knoten wird dann überwacht, wenn die Variablenwerte gelesen, und gesteuert, wenn die Variablenwerte geändert werden. Außer den Lese- und Schreib-Operationen werden im SNMP Operationen zur Durchquerung aller Variablen einer Liste oder Tabelle eines verwalteten Knotens, d.h. die Operation *get-next*, und zur Benachrichtigung der NMS durch den verwalteten Knoten wegen eines außergewöhnlichen Ereignisses, d.h. die Operation *Trap*, definiert.
Das SNMP dient also der Übertragung von Management-Informationen zwischen den Netzwerkmanagement-Stationen und den verwalteten Knoten. Der strukturelle Aufbau des Protokolls SNMP wird weiter unten in „3.6.3 Simple Network Management Protocol (SNMP)“ auf Seite 95 beschrieben.

### 3.4.4 Vertretungsagent

Jede Netzwerkmanagement-Architektur soll so entworfen sein, daß das Netzwerkmanagement allgegenwärtig *(ubiquitous)* ist, d.h. die Fähigkeit der Überwachung und Steuerung aller vorhandenen Geräte im Netzwerk besitzt. Das

---

13 siehe auch „3.6.3.3 Nachrichten“ auf Seite 98

14 siehe auch „3.6.3.2 Administrative Beziehungen“ auf Seite 97

15 siehe auch „3.6.3.6 Traps“ auf Seite 101

SNMP-Netzwerkmanagement basiert daher auf einem grundlegenden Prinzip, das auch das fundamentale Axiom *(fundamental axiom)* genannt wird, wodurch die Forderungen an die verwalteten Knoten möglichst gering bleiben müssen, damit viele Netzwerkgeräte überwacht und gesteuert werden können.
Außerdem stellt das SNMP-Netzwerkmanagement eine Möglichkeit zur Verfügung, zusätzlich zu den SNMP-unterstützenden verwalteten Knoten auch die sogenannten „fremden" Geräte im Netzwerk überwachen und steuern zu können. Unter fremden Geräten sollen die Netzwerkgeräte verstanden werden, die entweder das Netzwerkmanagement-Protokoll SNMP nicht unterstützen oder einen anderen Protokollsatz statt Internetprotokollsatz implementiert haben. Die fremden Geräte werden mittels eines speziellen Agenten, des Vertretungsagenten *(proxy agent)*, dem Management unterworfen, wobei NMS den Vertretungsagent für die Kommunikation anstelle des fremden Geräts adressiert (vgl. Abb. 26). Dabei besteht die Aufgabe des Vertretungsagenten darin, die von NMS an ihn gerichteten Protokollwechselwirkungen in Protokollwechselwirkungen, die das fremde Gerät versteht, zu übersetzen [Rose91a, Robe90, Step91, Mard91].

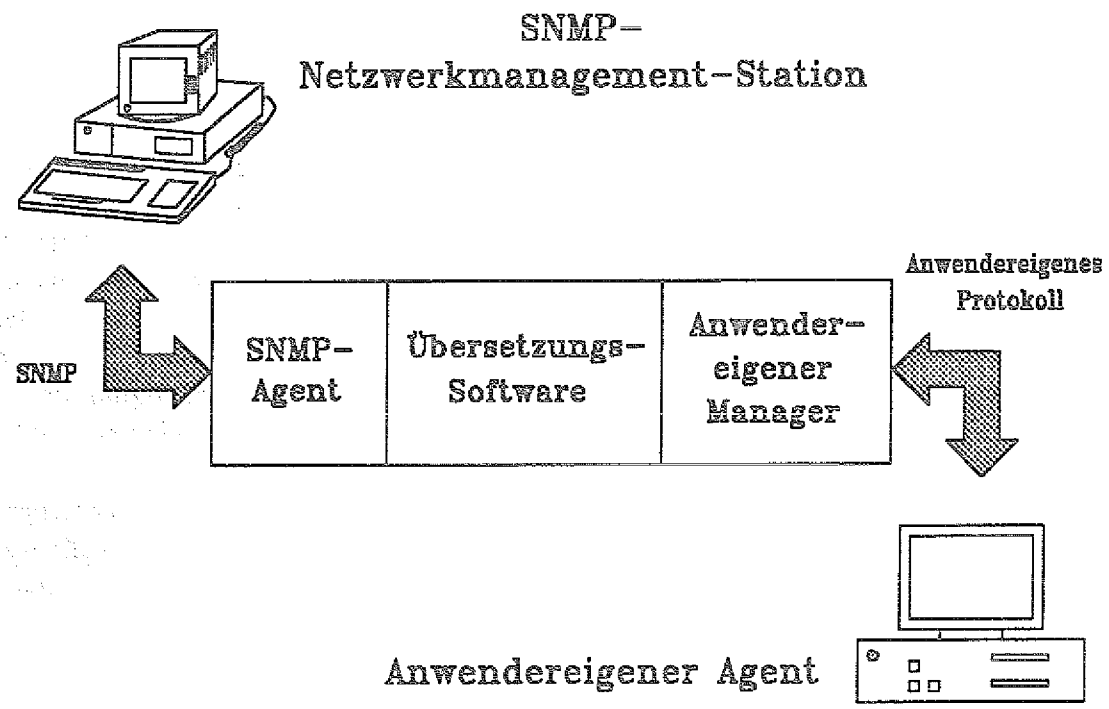



**Abb. 26. Vertretungsagent:** Das Konzept des Vertretungsagenten *(proxy agent)* erlaubt solche Geräte zu verwalten, die das Protokoll SNMP nicht implementiert haben.

Der Übersetzungsumfang wird durch das fremde Gerät bestimmt, d.h. wenn das fremde Gerät z.B. das Netzwerkmanagement-Protokoll SNMP unterstützt aber nicht die darunterliegenden Ende-zu-Ende-Dienste, wie z.B. UDP und IP zum Transport der SNMP-PDUs implementiert hat, dann braucht der Vertretungsagent nur die SNMP-PDUs aus dem empfangenen Datenpaket zu trennen und mit einem anderen Übertragungsdienst abzuschicken. Für den Fall, daß nur das Protokoll SNMP nicht

unterstützt wird, spielt ein Vertretungsagent die Rolle von einem Gateway für die Anwendungsschicht.

Der Vertretungsagent, der in der Regel ein Protokollumsetzer *(protocol converter)* ist, kann auch zwischen verwalteten Knoten einerseits und der NMS andererseits zur Aufbewahrung von Management-Informationen eingesetzt werden. So kann die Verarbeitungslast der verwalteten Objekte in Umgebungen, wo bestimmte Management-Anfragen häufig vorkommen und die Antworten sich nur selten ändern, reduziert werden.
Das Konzept des Vertretungsagenten erlaubt also das Management von anwendereigenen Umgebungen in das SNMP-Netzwerkmanagement einzubeziehen und dadurch das Management der realen Netzwerke zu erleichtern.

### 3.4.5 Charakteristische Eigenschaften des Modells

Das SNMP-Netzwerkmanagement-Modell minimiert die Anzahl und die Komplexität der von den Agenten auszuführenden Netzwerkmanagement-Funktionen. Dabei werden z.B. folgende Ziele erreicht:

- Die Kosten für die Entwicklung von Agent-Software werden reduziert;
- Die verwalteten Knoten werden von der Ausführung von Management-Aufgaben größtenteils befreit, d.h. es bleibt mehr Zeit für die eigentliche Funktionalität des Gerätes;
- Die Netzwerkmanagement-Werkzeuge werden wenigen Beschränkungen bezüglich der Form und der Komplexität unterworfen, da sie nach diesem Modell auf der NMS laufen und nicht mit jedem verwalteten Knoten kompatibel sein brauchen;
- Die vereinfachten Netzwerkmanagement-Funktionen sind leichter zu verstehen und werden deshalb bei der Entwicklung von Netzwerkmanagement-Werkzeugen bevorzugt verwendet [CFSD90a].

Außerdem ist das SNMP-Netzwerkmanagement-Modell durch die Erweiterbarkeit, die vor allem der Realisierung der unberücksichtigten Aspekte oder der Erfüllung der zukünftigen Forderungen an das Netzwerkmanagement dient, und die relative Unabhängigkeit von der Architektur oder den Mechanismen eines bestimmten Hosts oder Gateways gekennzeichnet.

## 3.5 Datendarstellung

Das SNMP-Netzwerkmanagement benutzt eine abstrakte Syntax *(abstract syntax)*, um die Abhängigkeit der Daten von ihrer internen Darstellung in jeder Maschine zu vermeiden. So können z.B. die Netzwerkmanagement-Informationen in Form einer unabhängigen abstrakten Syntax von einer NMS zu einem Agenten übertragen werden, obwohl die interne Darstellung der Daten in der NMS und im Agenten z.B. von der Programmiersprache und der Maschinenarchitektur abhängig ist. Um die so beschriebenen maschinenunabhängigen Daten eindeutig zu übertragen, wird außerdem eine Übertragungssyntax *(transfer syntax)* benötigt.

Im SNMP-Netzwerkmanagement werden Teile von ASN.1 [ISO8824] als eine maschinenunabhängige Sprache für Datenstrukturen (abstrakte Syntax) und BER [ISO8825] als Kodierungsregeln (Übertragungssyntax) verwendet. Einige wichtige Datenstrukturen, die für das Verständnis der späteren Darstellungen erforderlich sind und in SNMP-Netzwerkmanagement benutzt werden, sollen im folgenden kurz betrachtet werden [Rose91a, Rose90a, GoSp90]. Die Kodierungsregeln BER werden in dieser Arbeit nicht behandelt und sind in [Rose91a, ISO8825] beschrieben.

Unter dem Begriff ***Modul*** *(module)* versteht man eine Sammlung von ASN.1-Beschreibungen, die ein gemeinsames Thema betreffen, wie z.B. die Spezifikation eines Protokolls. Die Syntax eines Moduls wird im folgenden dargestellt:

```
<<Modul>> DEFINITIONS ::= BEGIN
<<Verbindung>>
<<Erklärung>>
END
```

Dabei bezeichnen die Ausdrücke

- < < *Modul* > > : den Modulnamen gefolgt vom OBJECT IDENTIFIER, der weiter unten beschrieben ist. Die Angabe des OBJECT IDENTIFIER ist in diesem Fall wahlfrei;
- < < *Verbindung* > > : die Objektdefinitionen, die andere Module diesem Modul zur Verfügung stellen *(IMPORTS)*, und die Objektdefinitionen, die dieses Modul den anderen Modulen bereitstellt *(EXPORTS)*;
- < < *Erklärungen* > > : die eigentliche Definition in ASN.1-Notation.

Drei mittels ASN.1-Sprache definierte Objektarten, werden im folgenden beschrieben (vgl. Abb. 28):

- ***Typen*** *(types)* definieren eine neue Datenstruktur und besitzen die folgende Syntax:

  ```
  NameOfType ::=
      TYPE
  ```

  Der Name eines Types *(NameOfType)* ist ein ASN.1-Wort[16], wobei nach ASN.1-Konvention der erste Buchstabe groß geschrieben wird, wie z.B. *Counter*.

- ***Werte*** *(values)* sind die Instanzen bzw. Variablen eines Typs und werden wie folgt syntaktisch definiert:

  ```
  nameOfValue NameOfType ::=
      VALUE
  ```

---

16 Ein ASN.1-Wort besteht aus Bindestrich, Groß- und Kleinbuchstaben. Alle Buchstaben der ASN.1-Schlüsselwörter werden groß geschrieben [Rose90a].

In der obigen Zuweisung wird zuerst der Name des Wertes *nameOfValue* und dann der Objekttyp, dessen Instanz dieser Wert ist, angegeben. Der Name eines Wertes ist nach ASN.1-Konvention ein ASN.1-Wort, das mit einem Kleinbuchstaben anfängt, wie z.B. *directory*.

- *Makros (macros)* erlauben, die ASN.1-Grammatik modifiziert zu benutzen. Im SNMP-Netzwerkmanagement wird nur ein ASN.1-Makro, nämlich der OBJECT-TYPE-Makro definiert, der zur Definition von verwalteten Objekten benutzt wird. Die Syntax dieses speziellen Makros wird daher in „3.6.1.3 Verwaltete Objekte" auf Seite 82 angegeben. Hier soll noch darauf hingewiesen werden, daß der Name eines Makros nach ASN.1-Konvention ein ASN.1-Wort ist und nur aus Großbuchstaben besteht, wie die ASN.1-Schlüsselwörter.

Das SNMP-Netzwerkmanagement benutzt vier Arten von ASN.1-Typen:

*Einfache Typen (simple types)*

- INTEGER: stellt einen Datentyp dar, dessen Wert eine ganze Zahl ist. Zusätzlich zu den ganzen Zahlen können den Werten eines INTEGER-Typs symbolische sinnvolle Namen zugeordnet werden, die einer besseren Lesbarkeit dienen.

- OCTET STRING: bezeichnet einen Datentyp, der eine Kette von Oktetten als Wert hat. Jedes Oktett in einem OCTET STRING kann die Werte zwischen 0 und 255 haben.

- OBJECT IDENTIFIER: ist ein Datentyp, der die vorgeschriebene Benennung eines Objekts angibt und deshalb Objekte unabhängig von der mit ihnen verbundenen Semantik identifizieren kann, wie z.B. ein Dokument oder ein ASN.1-Modul. Der Wert des OBJECT IDENTIFIER ist eine Folge von nicht-negativen ganzen Zahlen, die gewonnen werden, wenn im Namensbaum vom *root* bis zum gesuchten Objekt die auf dem Weg liegenden und mit den Knoten verbundenen Zahlen genommen werden. Die Zahlen in der Zahlenfolge werden in der kurzgefaßten Benennung durch Punkte voneinander getrennt, die im folgenden für den *mib(1)* dargestellt ist (vgl. Abb. 27):

  ```
  1.3.6.1.1
  ```

  Die Baumknoten besitzen zusätzlich zu den numerischen Werten auch eindeutige Namen in Textform, die beim Durchqueren des Baums die administrative Kontrolle des betreffenden Knoten anzeigen. Eine Kombination der numerischen und textlichen Werte sorgt für eine bessere Lesbarkeit und wird deshalb häufig benutzt. Zusätzlich zu der kurzgefaßten Benennung kann *mib(1)* folgendermaßen dargestellt werden[17]:

---

17 In der dritten Angabe wurde der Name *1.3.6.1.2* durch seinen symbolischen Namen *mgmt* ersetzt.

```
mib(1)   OBJECT IDENTIFIER ::= { 1 3 6 1 2 1 }
mib(1)   OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 2 1 }
mib(1)   OBJECT IDENTIFIER ::= { mgmt 1 }
```

- NULL: kennzeichnet einen selten im SNMP-Netzwerkmanagement benutzten Datentyp, der als Platzhalter wirkt.

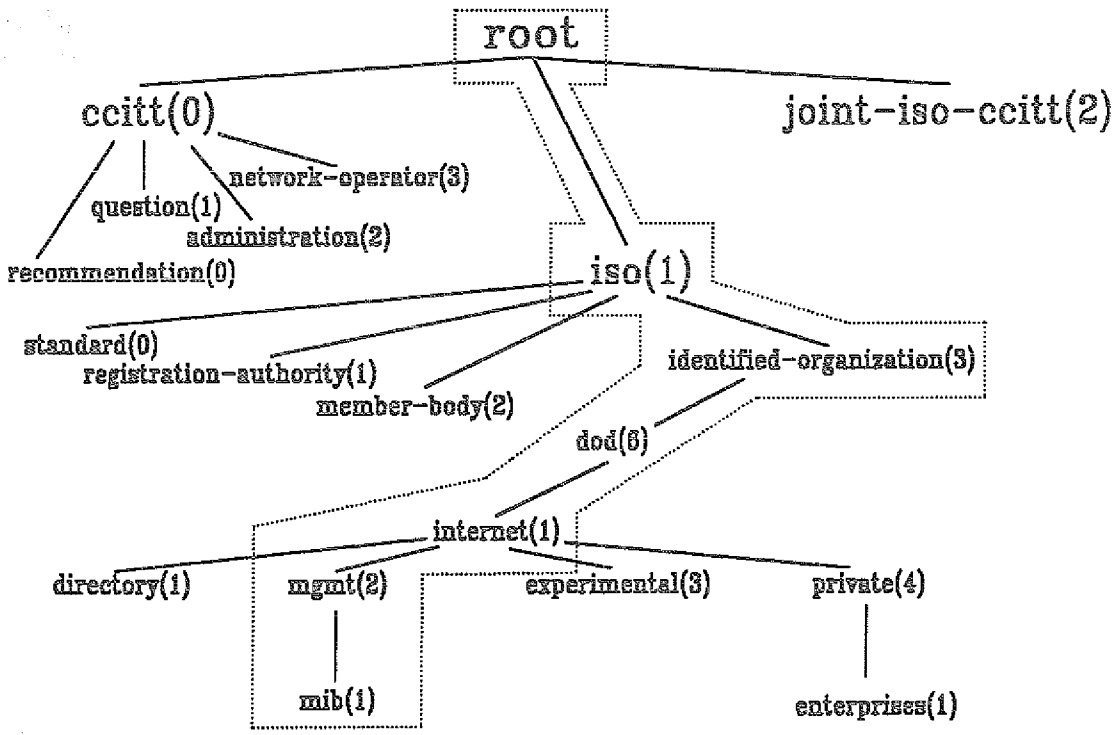



**Abb. 27.** Objektbezeichnerbaum: Ein Teil des Objektbezeichnerbaumes, der beliebig tief fortsetzbar zu denken ist, wird abgebildet, um den Mechanismus zur Benennung der Objekte mittels des Datentyps OBJECT IDENTIFIER zu erläutern. Der *root* besitzt keine numerische und textliche Kennung. In der Abbildung wurde beispielsweise durch die punktierte geschlossene Linie der Weg, welcher der Benennung von *mib(1)* dient, vom *root* bis zum Knoten *mib(1)* gekennzeichnet. Alle Knoten des Objektbezeichnerbaumes besitzen eine numerische und eine textliche Kennung, wobei die Angabe der numerischen Bezeichnung ausreichend ist. Die Benutzung der textlichen Angaben ist nur erlaubt, wenn der Nachfolger eines Knotens eindeutig den Knoten kennzeichnet.

*Konstruierte Typen (constructed types)*

- SEQUENCE: stellt eine geordnete Liste von Null oder mehr unterschiedlichen ASN.1-Typen dar.

- SEQUENCE OF: ist im Unterschied zu SEQUENCE eine geordnete Liste von Null oder mehr gleichen ASN.1-Typen.

*Etikettierte Typen (tagged types)*

ASN.1 ordnet jedem Datentyp ein Etikett *(tag)* zu, mit dem die ASN.1-Typen zur besseren Unterscheidung in vier verschiedene Klassen unterteilt werden:

- Universaletikett *(universal tag)*: spezifiziert einen Datentyp, der global eindeutig ist;
- Anwendungsetikett *(application-wide tag)*: bezeichnet einen Datentyp, der nur innerhalb eines ASN.1-Moduls eindeutig sein soll;
- Kontextetikett *(context-specific tag)*: spezifiziert einen Datentyp, der nur innerhalb eines konstruierten Typs, wie z.B. SEQUENCE, eindeutig sein muß. Das Kontextetikett wird im SNMP-Netzwerkmanagement nicht benutzt;
- Privatetikett *(private-use tag)*: erlaubt Datentypen zu definieren, die nur innerhalb eines privaten Bereiches, wie z.B. eines Unternehmens, eindeutig sein brauchen.

Zur Definition eines neuen Datentyps mittels Etiketten wird nach dem Namen und dem Zuweisungssymbol das Etikett angegeben, gefolgt von einer nicht-negativen Nummer in eckigen Klammern, die der Unterscheidung der gleichen etikettierten Datentypen dient. Ausnahme macht das Kontextetikett dabei in der Art, daß das Kontextetikett nach ASN.1-Konvention mit jedem Element des konstruierten Typs benutzt werden darf, während die anderen Etiketten nur am Anfang einer Typbeschreibung stehen dürfen[18]. Als Beispiel wird die Definition eines Datentyps IA5String, der das Universaletikett mit der Nummer 22 besitzt, angegeben:

```
IA5String ::=
    [UNIVERSAL 22]
        IMPLICIT OCTET STRING
```

Die Angabe IMPLICIT ist ein Hinweis für die Übertragungssyntax und bedeutet, daß keine Informationen zur Übertragung auf dem Netz dem neuen Datentyp eingefügt werden müssen. Aus diesem Grund wird nur das mit IA5String verbundene Etikett, d.h. das Universaletikett mit der Nummer 22, und nicht das mit dem OCTET STRING verbundene Etikett auf dem Netz übertragen [Rose90a].

*Modifizierte Typen (Subtypes)*

Die Möglichkeit, die existierenden Datentypen zu modifizieren, erlaubt es, verfeinerte Datentypen zu definieren. Im SNMP-Netzwerkmanagement werden nur zwei Formen der Verfeinerung verwendet, die durch die weiter unten beschriebenen Beispiele erläutert werden [Rose91a]:

```
IpAddress ::=        -- nach Netzwerk Byte-Ordnung
    [APPLICATION 0]
        IMPLICIT OCTET STRING (SIZE (4))
```

---

18 siehe [Rose90a], Seite 249

In diesem Fall wird ein neuer Datentyp IpAddress definiert, der aus eine Oktettkette der Länge vier besteht. Außerdem zeigt das Beispiel an, daß Kommentare nach dem Symbol '--' eingefügt werden dürfen.

```
Gauge ::=
    [APPLICATION 2]
        IMPLICIT INTEGER (0..4294967295)
```

Dieses Beispiel zeigt die Definition eines neuen Datentyps, der nur ganzzahlige Werte kleiner als $2^{32}$ haben kann.

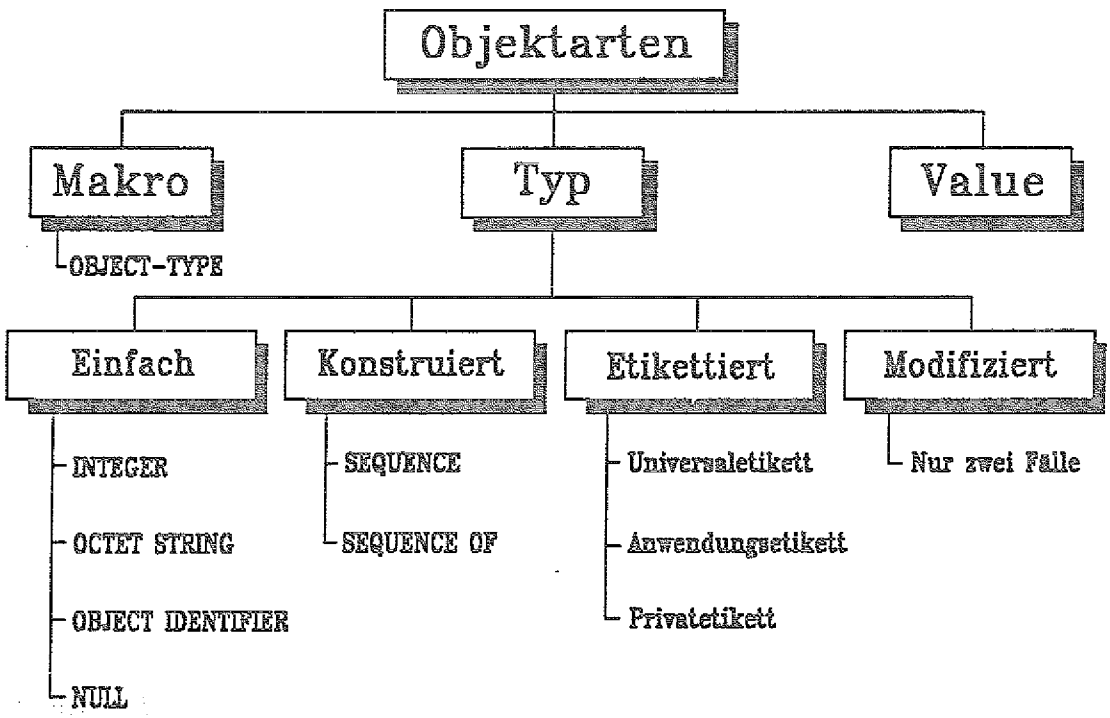



Abb. 28. Objektarten im SNMP-Netzwerkmanagement: Dargestellt ist eine Übersicht von im SNMP-Netzwerkmanagement verwendeten Objektarten.

## 3.6 SNMP-Netzwerkmanagement-Rahmenwerk

Das SNMP-Netzwerkmanagement-Rahmenwerk wird durch die drei folgenden Dokumente, die die Standardisierungsstufe *Standard* im IAB erreicht haben, repräsentiert:

- *Structure and Identification of the Management Information for TCP/IP-based internets, SMI, RFC1155* [RoMc90a],
- *Management Information Base for Network Management of TCP/IP-based internets, MIB, RFC1156* [McRo90], der auch MIB-I genannt wird, und
- *Simple Network Management Protocol, SNMP, RFC1157* [CFSD90a].

Im weiteren Verlauf dieses Kapitels wird das SNMP-Netzwerkmanagement auf diesem Rahmenwerk aufbauend erläutert.

### 3.6.1 Struktur von Management-Informationen (SMI)

Management-Informationen werden im SNMP-Netzwerkmanagement in einem virtuellen Speicher, der MIB genannt wird und in der Regel durch eine Datenbank realisiert wird, aufbewahrt. Die Struktur der Management-Informationen wird in der SMI *(Structure of Management Information)* beschrieben. Dabei wird eine Menge von Regeln spezifiziert, die beschreibt, wie Netzwerkobjekte für die Benutzung durch das Protokoll definiert werden, wie das Protokoll auf die Objekte zugreifen darf und wie Objekte der MIB zugefügt werden. Die SMI benutzt nur eine Teilmenge der ASN.1-Notation, wie im Abschnitt „3.5 Datendarstellung" auf Seite 73 erläutert wurde. Diese Begrenzung dient der Einfachheit *(simplicity)* des SNMP-Netzwerkmanagements.
Die SMI wird durch die Merkmale Einfachheit und Erweiterbarkeit *(extensibility)* gekennzeichnet. Einfachheit sorgt für die minimale Begrenzung bezüglich Kompatibilität, während Erweiterbarkeit die Anpassung an die neuen Anforderungen ermöglicht [Kral90, Rose91a, RoMc90a, Robe90]. Die wesentlichen Definitionen und Komponenten der SMI werden in den nächsten Abschnitten beschrieben.

#### 3.6.1.1 Namen

Zur Kennzeichnung von verwalteten Objekten werden Namen benutzt, die einen hierarchischen Aufbau besitzen. Dabei wird der Datentyp OBJECT IDENTIFIER verwendet, der auch für andere Zwecke benutzt wird, wie z.B. zur Bezeichnung eines internationalen Standards [Benn90]. Die jedem Knoten des Objektbezeichnerbaums zugeordneten textlichen Angaben, wie in „3.5 Datendarstellung" auf Seite 73 beschrieben wurde, lassen auf die für die Verwaltung des jeweiligen Knoten zuständige Instanz zurückschließen. Beispielsweise bestimmt der Internet-Knoten, der durch IAB verwaltet wird, unter welcher Strategie der Teilbaum verwaltet wird. Der Internet-Knoten wird wie folgt bezeichnet (siehe auch Abb. 27):

```
internet      OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }
```

Im Internet-Teilbaum werden vier verschiedene Objektklassen oder Teilbäume definiert[19]:

```
directory     OBJECT IDENTIFIER ::= { internet 1 }
mgmt          OBJECT IDENTIFIER ::= { internet 2 }
experimental  OBJECT IDENTIFIER ::= { internet 3 }
private       OBJECT IDENTIFIER ::= { internet 4 }
```

- *Verzeichnisklasse (directory)*: ist für zukünftigen Gebrauch reserviert;

---

[19] Das Wort *internet* in geschweiften Klammern deutet darauf hin, daß die entsprechende Objektgruppe mit dem Präfix *1.3.6.1* anfängt.

- *Managementklasse (mgmt)*: enthält Objekte, die innerhalb von IAB gebilligten Dokumenten definiert werden, z.B. MIB-I (*RFC1156*, [McRo90]) und MIB-II (*RFC1158 veraltet, RFC1213*, [Rose90c, Rose91d]);

- *Experimentklasse (experimental)*: ist für die Objekte vorgesehen, die in der Testphase sind;

- *Privatklasse (private)*: erlaubt die Definition von willkürlichen Objekten, wie z.B. unternehmensspezifischen Objekten.

Nur die drei letzten Klassen, d.h. Management-, Experiment- und Privatklasse werden im SNMP-Netzwerkmanagement betrachtet, und ihre Objekte werden durch IANA *(Internet Assigned Numbers Authority)* registriert.

#### 3.6.1.2 Syntax

Die in der ASN.1-Sprache dargestellte Syntax der Objekttypen, die im SNMP-Netzwerkmanagement verwendet werden, sind in der SMI beschrieben. Dabei werden die drei folgenden Klassen unterschieden:

- Einfache Typen, wie INTEGER, OCTET STRING, OBJECT IDENTIFIER und NULL, die in dieser Arbeit in „3.5 Datendarstellung" auf Seite 73 beschrieben wurden. Hier soll noch darauf hingewiesen werden, daß die SMI die Zuordnung vom Wert Null zu einem symbolischen Namen in einer INTEGER-Aufzählung[20] nicht erlaubt. Eine weitere Konvention, die aber in der SMI nicht festgelegt ist, beschränkt die Menge der Werte, die in einer Aufzählung den Instanzen eines Datentyps zur Verfügung stehen, auf die in der Aufzählung angegebenen Werte. Außerdem muß die in einem OCTET STRING für die Oktettkette benutzte Repertoire entweder binär oder der NVT-ASCII-Zeichensatz[21] sein. Auch diese Konvention wird in der SMI nicht beschrieben [Rose91a].

- Sechs neue Datentypen, d.h. *NetworkAddress*, *IpAddress*, *Counter*, *Gauge*, *TimeTicks* und *Opaque*, die für die Benutzung im SNMP-Netzwerkmanagement definiert sind:

  1. *NetworkAddress*: stellt eine Auswahl von Netzwerkadressen in verschiedenen Protokollfamilien zur Verfügung, obwohl zur Zeit nur eine Protokollfamilie, d.h. der Internet-Protokollsatz definiert worden ist. Die Syntax lautet:

```
NetworkAddress ::=
    CHOICE {
        internet
            IpAddress
    }
```

---

20 siehe z.B. die INTEGER-Aufzählung in „Anhang C. TOASTER-MIB" auf Seite 155

21 NVT-ASCII-Zeichensatz *(Network Virtual Terminal ASCII character set)* ist auf Seite 10 in [Post83] beschrieben worden.

2. *IpAddress*: repräsentiert eine IP-Adresse, d.h. eine 32-Bit-Internet-Adresse, und ist wie folgt definiert:

```
IpAddress ::=        -- Nach Netzwerk Byte-Ordnung
    [APPLICATION 0]
        IMPLICIT OCTET STRING (SIZE (4))
```

3. *Counter*: stellt einen nicht-negativen ganzzahligen Wert dar, der monoton bis zu einem maximalen Wert hochzählt und nach Überschreitung einer Grenze wieder bei Null anfängt. Syntaktisch wird der Datentyp *Counter* beschrieben durch:

```
Counter ::=
    [APPLICATION 1]
        IMPLICIT INTEGER  (0..4294967295)
```

4. *Gauge*: ist ein nicht-negativer ganzzahliger Wert, der durch einen maximalen Wert begrenzt ist. Im Unterschied zu *Counter* darf *Gauge* ab- und zunehmen; er wird wie folgt dargestellt:

```
Gauge ::=
    [APPLICATION 2]
        IMPLICIT INTEGER  (0..4294967295)
```

5. *TimeTicks*: ist ein nicht-negativer ganzzahliger Wert, welcher die Zeit in Hundertstelsekunden ab einem Anfangszeitpunkt, der bei der Definition eines Objekts in der MIB festgelegt wird, angibt. Die Syntax lautet:

```
TimeTicks  ::=
    [APPLICATION 3]
        IMPLICIT INTEGER  (0..4294967295)
```

6. *Opaque*: kann jeden willkürlichen ASN.1-Datentyp repräsentieren, der nach Kodierungsregeln BER *(Basic Encoding Rules)* kodiert worden ist und in Form einer Oktettkette, die den Wert von *Opaque* darstellt, vorliegt. So wird der ursprüngliche ASN.1-Datentyp zweimal kodiert *(double-wrapping)*, erstens durch die BER-Kodierung in Form einer Oktettkette und zweitens durch die Zuordnung zu *Opaque* in Form von OCTET STRING. Eine konforme Implementation braucht nur die zweimal kodierten Daten zu erkennen und zu akzeptieren, wobei die Dekodierung und Interpretation von Daten nicht vorgeschrieben wird. *Opaque* hat die folgende Syntax:

```
Opaque  ::=          -- Willkürliche ASN.1-Wert,
    [APPLICATION 4]  -- "zweimal kodiert"
        IMPLICIT OCTET STRING
```

- Konstruierte Typen, wie SEQUENCE und SEQUENCE OF, die zur Definition der folgenden zwei Datentypen benutzt werden:

1. *Liste (list)*: enthält eine Folge von oben genannten einfachen und etikettierten Datentypen, die in folgender Syntax durch <*type1*> bis <*typeN*> gekenn-

zeichnet sind. Im Unterschied zu ASN.1 ist die Benutzung von Ausdrücken OPTIONAL und DEFAULT in der SMI nicht erlaubt. Eine Liste wird syntaktisch wie folgt aufgebaut:

```
<list>  ::=
    SEQUENCE {
        <type1>,
        ...
        <typeN>
    }
```

2. ***Tabelle** (table)* setzt sich aus eine Folge von Listen zusammen. So ist eine Tabelle im SNMP-Netzwerkmanagement zweidimensional und besteht aus Null oder mehreren Reihen, die jeweils die gleiche Anzahl von Spalten haben. Ihre Syntax lautet:

```
<table>  ::=
    SEQUENCE OF
        <list>
```

#### 3.6.1.3 Verwaltete Objekte

Zur Beschreibung von verwalteten Objekten wird das in der SMI definierten OBJECT-TYPE-Makro verwendet, dessen Syntax gegeben ist durch[22]:

```
OBJECT-TYPE MACRO ::=
BEGIN
    TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax)
                      "ACCESS" Access
                      "STATUS" Status
    VALUE NOTATION ::= value (VALUE ObjectName)

    Access ::= "read-only"
                    | "read-write"
                    | "write-only"
                    | "not-accessible"
    Status ::= "mandatory"
                    | "optional"
                    | "obsolete"
END
```

22 In der Syntax deutet eine in "" eingeschlossene Zeichenkette auf den mit der Zeichenkette verbundenen Wert hin.

Dabei spezifizieren die Ausdrücke

- *SYNTAX*: den das Objekt modellierenden Datentyp, welcher hier durch *ObjectSyntax* gekennzeichnet ist und jeder in der SMI definierter Datentyp sein kann[23];
- *ACCESS*: die erlaubte Zugriffsart, d.h. Nur-Lesen *(read-only)*, Lesen-Schreiben *(read-write)*, Nur-Schreiben *(write-only)* oder unzugänglich *(not-accessible)*;
- *STATUS*: die Implementationsforderung für die verwalteten Objekte, wie
  - „obligatorisch *(mandatory)*: Verwalteter Knoten muß das Objekt implementieren,
  - wahlfrei *(optional)*: Verwalteter Knoten darf das Objekt implementieren und
  - veraltet *(obsolete)*: Verwalteter Knoten braucht das Objekt nicht mehr zu implementieren"[24];
- *VALUE*: den Objektnamen durch den Datentyp *ObjectName*, der als OBJECT IDENTIFIER definiert wird:

```
ObjectName ::=
    OBJECT IDENTIFIER .
```

Die syntaktische Darstellung des verwalteten Objekts *sysUpTime*[25] ist weiter unten als Beispiel angegeben. Dabei ist das Objekt vom Datentyp *TimeTicks*, seine Werte können nur gelesen werden, seine Implementation in einem verwalteten Knoten ist obligatorisch und schließlich ist sein Name durch den OBJECT IDENTIFIER { *system 3* } festgelegt.

```
sysUpTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        ::= { system 3 }
```

Während ein Objekttyp die Art eines verwalteten Objekts definiert, wie z.B. das Format eines Eintrags in einer Tabelle, spezifiziert eine Objektinstanz einen bestimmten Eintrag in der Tabelle und ist mit einem Wert verbunden. Die Konventionen zur Bezeichnung von Objektinstanzen sind in [CFSD90a, Rose90c, Rose91d] angegeben.

---

23 Die Syntax von Datentyp *ObjectSyntax* ist im SMI-Modul [RoMc90a] auf Seiten 17-18 angegeben.

24 siehe Seite 94 in [Rose91a]

25 Der textliche Name eines Objekts wird OBJECT DESCRIPTOR genannt [ISO8824].

### 3.6.2 Management-Informationsbasis (MIB)

Im SNMP-Netzwerkmanagement werden die verwalteten Objekte über einen virtuellen Informationsspeicher, der Management-Informationsbasis *(Management Information Base, MIB)*, überwacht und gesteuert. Die Objekte in der MIB, deren Struktur mittels einer Teilmenge der ASN.1-Sprache in der SMI beschrieben wird, ermöglichen einen einheitlichen protokollunabhängigen Datenraum. Die MIB ist konzeptionell in Form eines Baumes mit Knoten, die mit symbolischen Namen gekennzeichnet sind, organisiert, wobei die Objekte die Blätter dieses Baumes repräsentieren. Die Objekte werden dann eindeutig identifiziert, indem der Weg vom *root* bis zum jeweiligen Objekt angegeben wird (siehe auch Abb. 27).

Die Objekte in einer MIB können im SNMP-Netzwerkmanagement vier verschiedenen Klassen angehören, wobei im Rahmen des Netzwerkmanagements drei Klassen (Management-, Experiment- und Privatklasse) betrachtet werden. Objekte der Managementklasse, die das Präfix *mib = 1.3.6.1.2.1* im Objektbezeichnerbaum besitzen, sind standardisiert und werden deshalb im folgenden erläutert.

Das zweite Dokument im SNMP-Netzwerkmanagement-Rahmenwerk, d.h. MIB-I [McRo90], definiert eine minimale Anzahl von Objekten, die gezielt ausgewählt worden sind, um die Verwaltung eines den Internet-Protokollsatz implementierenden Knotens zu sichern. Für die Auswahl der Objekte sind beispielsweise folgende Bedingungen berücksichtigt worden:

- Die Anzahl der Objekte muß auf ca. 100 begrenzt bleiben;
- Die Objekte müssen hauptsächlich der Konfigurations- und Fehleranalyse dienen;
- Zur Vermeidung von Redundanz darf ein Objekt nicht auf eine einfache Weise von einem anderen ableitbar sein.

MIB-I wird in funktionale Gruppen unterteilt, die in einem Netzwerkgerät implementiert sein müssen, wenn das Gerät als verwalteter Knoten gelten soll. Abhängig davon, inwieweit der Internet-Protokollsatz durch den Knoten unterstützt wird, müssen entweder alle oder einige dieser funktionalen Gruppen in verwalteten Knoten implementiert werden. Beispielsweise muß ein Gateway, der das Protokoll EGP *(Exterior Gateway Protocol)* implementiert, die weiter unten beschriebene EGP-Gruppe implementieren, während die Implementation der EGP-Gruppe für Hosts nicht obligatorisch ist. Dabei ist zu berücksichtigen, daß im SNMP-Netzwerkmanagement die Implementation eines Objekts einer Gruppe die Implementation der ganzen Gruppe erzwingt. So besitzt jeder Agent seine MIB, die mindestens die für den verwalteten Knoten obligatorischen funktionalen Gruppen enthält. Der Agent hat dann bezüglich seiner MIB zwei Funktionen auszuführen [Sabo91]:

1. Prüfung und Abfrage der Instanzen von Objekten, auch Variablen genannt („3.4.2 Verwaltete Knoten“ auf Seite 70), und
2. Änderung der Variablen.

Durch die Abfrage und Änderung der Management-Informationen in der MIB, d.h. Instanzen der Objekte oder Variablen, wird erst das Überwachen und Steuern von verwalteten Knoten möglich. Außer der in jedem verwalteten Knoten vorhandenen MIB existiert die NSD *(Network Statistics Database)*, die zentral der NMS zur Tendenzanalyse zur Verfügung steht.

Die Beschränkung der MIB-I auf ca. 100 Objekte ermöglicht eine einfache Implementierung der MIB und erlaubt, daß viele Netzkomponenten recht schnell überwacht und gesteuert werden können. Um aber die neuen oder modifizierten Komponenten zukünftig auch noch verwalten zu können, wurde in der SMI die Erweiterung der MIB zugelassen (MIB-II), was in Abschnitt „3.6.2.5 Erweiterungsmöglichkeiten" auf Seite 93 näher erläutert wird.

### 3.6.2.1 Aufbau der MIB-I und MIB-II

Die Objekte in MIB-I und MIB-II werden mittels des OBJECT-TYPE-Makros, das in der SMI beschrieben worden ist, definiert und sind in sogenannten funktionalen Gruppen organisiert (vgl. Tab. 5).

| Funktionalgruppe | Objektanzahl in MIB-I | Objektanzahl in MIB-II |
|---|---|---|
| Systemgruppe | 3 | 7 |
| Schnittstellengruppe | 22 | 23 |
| AT-Gruppe | 3 | 3 |
| IP-Gruppe | 33 | 38 |
| ICMP-Gruppe | 26 | 26 |
| TCP-Gruppe | 17 | 19 |
| UDP-Gruppe | 4 | 7 |
| EGP-Gruppe | 6 | 18 |
| Übertragungsgruppe | - | 0 |
| SNMP-Gruppe | - | 30 |
| *Summe* | 114 | 171 |

**Tab. 5. Objektanzahl in MIB-I und MIB-II:** MIB-II ist die neue Version der Internet-Standard-MIB (MIB-I) und umfaßt, zusätzlich zu den im MIB-I definierten Objekten, neue Objekte und Tabellen. Die gemeinsamen Objekte in MIB-I und MIB-II besitzen den gleichen Namen, da MIB-II das gleiche Namensschema wie MIB-I benutzt und als eine übergeordnete Objektmenge von MIB-I zu betrachten ist.

Die Definition der Objekte erfolgt nach einem einheitlichen Schema, das sich aus fünf Angaben zusammensetzt [Rose90c, Rose91d]:

1. der textliche Name des Objekts *(OBJECT DESCRIPTOR)* gefolgt von Objektbezeichner *(OBJECT IDENTIFIER)*,

2. die abstrakte Syntax für das Objekt aus der Menge der in der SMI innerhalb *ObjectSyntax* definierten Datentypen,

3. eine textliche Beschreibung der Semantik des Objekts, die für eine einheitliche Interpretation des Objekttyps in allen Umgebungen sorgt,

4. die Zugriffsart *(Access)* und

5. die Implementierungsforderung für das Objekt *(Status)*, wie in „3.6.1.3 Verwaltete Objekte“ auf Seite 82 beschrieben worden ist.

### 3.6.2.2 Konventionen

Die Namen der Objekte in der MIB-I und MIB-II sind drei Konventionen unterworfen, wobei die dritte Konvention nicht in den MIB-Dokumenten beschrieben wird:

1. Kein Name eines Objekts darf mit Null enden, da dieser Wert für den zukünftigen Gebrauch reserviert worden ist. Daher wird das erste Objekt in der IP-Gruppe, *ipForwarding*, folgendermaßen beschrieben:

```
iso  org  dod  internet  mgmt  mib  ip  ipForwarding
 1    3    6       1       2    1    4        1

ipForwarding = 1.3.6.1.2.1.4.1 = ip 1 .
```

2. Der textliche Name, d.h. der OBJECT DESCRIPTOR, muß möglichst kurz und aus mnemotechnischen Zeichenketten zusammengesetzt sein, wie z.B.

```
ipAdEntBcastAddr = 1.3.6.1.2.1.4.20.1.4 .
```

3. Der textliche Name eines Objekts, dessen Syntax der Datentyp *Counter* ist, muß mit dem Buchstaben "s" enden, wie die folgenden Beispiele verdeutlichen.

```
ipInReceives = ip 3
ifOutNUcastPkts = ifEntry 18
snmpInTooBigs = snmp 8
```

Alle Objekte, die als Internet-Standard-MIB definiert werden, d.h. MIB-I und MIB-II, dürfen nicht als Implementationsforderung den Zustand *(Status)* wahlfrei *(optional)* besitzen und werden gemäß Konvention als obligatorisch *(mandatory)* definiert. Im Gegensatz dazu werden die Objekte in der Experimentalklasse mit dem Zustand wahlfrei definiert und werden erst bei der Aufnahme in Internet-Standard-MIB ihren Zustand zu obligatorisch ändern.

### 3.6.2.3 Merkmale der MIB-II

Die MIB-II, die sich gegenwärtig im Standardisierungszustand *Proposed Standard* befindet, stellt eine neue Version für die MIB-I dar, welche ein Internet-Standard ist. Die MIB-II besitzt neue Objekte und Gruppen und ist mit der MIB-I und der SMI kompatibel. Die Erweiterungen können im allgemeinen aus drei Gründen erfolgen:

1. Ergänzung aufgrund neuer Betriebsforderungen,

2. verbesserte Unterstützung von Geräten, die verschiedene Protokolle unterstützen, und

3. textliche Verbesserungen.

In der MIB-II wird die Bezeichnung "mißbilligt" *(deprecated)* in der Definition von Objekten benutzt, um darauf hinzuweisen, daß das jeweilige Objekt sehr wahrscheinlich ab der nächsten Version der MIB nicht mehr verfügbar sein wird. Ein Beispiel hierfür stellt die *atTable* in der MIB-II dar, die als mißbilligt bezeichnet worden ist. Ihre Entfernung hat die Entfernung der gesamten AT-Gruppe zur Folge hat. Die AT-Gruppe, die die Abbildung von IP-Adressen auf physikalische Adressen ermöglicht, ist erfahrungsgemäß nur in Einprotokollumgebungen und nur für Abbildungen in eine Richtung effizient implementiert worden [Rose90c]. So wurden z.B. die Knoten nicht unterstützt, die das ES-IS-Protokoll[26] implementiert haben, und daher eine Abbildung in beiden Richtungen benötigen. Dieses Problem wird in der MIB-II dadurch gelöst, daß in der IP-Gruppe anstelle der als mißbilligt bezeichnete *atTable* für jede Protokollfamilie eine oder zwei Tabellen definiert werden, die die Abbildung in die jeweilige Richtung ermöglichen.
Im Falle des Internetprotokollsatzes ist die Tabelle *ipNetToMediaTable*, die die gleiche Funktionalität wie die *atTable* besitzt, in der IP-Gruppe definiert worden. Werden aber die Objekte für das CLNP[27] *(connectionless network protocol)* definiert, so werden sie zwei Tabellen für Abbildungen in beiden Richtungen enthalten.

Eine neue textliche Konvention, d.h. die Definition eines neuen Datentyps, ist in der MIB-II eingeführt worden, die es erlaubt, Mißdeutungen der MIB bezüglich des Datentyps OCTET STRING zu vermeiden. Der neu definierte Datentyp wird *DisplayString* genannt und ist auf den NVT-ASCII-Zeichensatz beschränkt. So stellt *DisplayString* mit der weiter unten dargestellten Syntax nur ein Mittel zur Präsentation dar und hat keinen Einfluß auf die Kodierung der Objekte, die mittels OCTET STRING definiert worden sind.

```
DisplayString ::=
    OCTET STRING
```

Beispiele hierfür sind die im folgenden angegebenen Objekte, die in der MIB-I syntaktisch mittels OCTET STRING definiert wurden und in der MIB-II die Syntax *DisplayString* besitzen.

```
sysDesc = system 1
ifDescr = 1.3.6.1.2.1.2.2.1.2
```

---

26 Endsystem zu Zwischensystem Protokoll *(End System to Intermediate System Protocol, ES-IS-Protokoll)* kennzeichnet ein OSI-Protokoll, das der Entdeckung von Zwischensystemen, wie z.B. Gateways, und Adreßbestimmung dient [Jaco91, Rose91a].

27 Ein OSI-Protokoll, das der Bereitstellung von verbindungslosen Vermittlungsdiensten, Datagramm-Diensten, dienen.

### 3.6.2.4 Funktionale Gruppen der MIB-I und MIB-II

Die innerhalb der MIB-I und MIB-II definierten Objekte stellen nur die für das Netzwerkmanagement notwendigen Objekte dar und sind daher keine wahlfreie Objekte. Die MIB-I organisiert die Objekte in acht funktionalen Gruppen:

1. Systemgruppe,
2. Schnittstellengruppe,
3. Adressenumwandlungsgruppe,
4. IP-Gruppe,
5. ICMP-Gruppe,
6. TCP-Gruppe,
7. UDP-Gruppe und
8. EGP-Gruppe.

Die obigen Gruppen sind um die zwei folgenden Gruppen, die in der MIB-II definiert worden sind, erweitert worden:

1. Übertragungsgruppe und
2. SNMP-Gruppe.

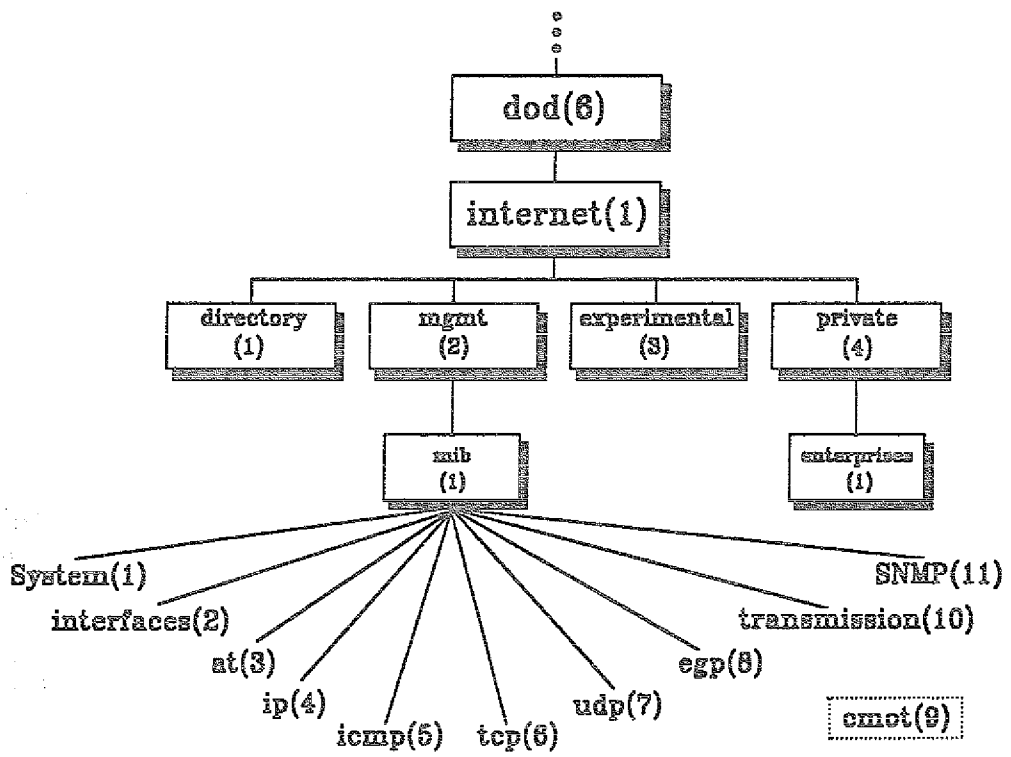



**Abb. 29. Funktionale Gruppen in MIB-I und MIB-II:** Abgebildet ist die Lage der funktionalen Gruppen im Objektbezeichnerbaum. Da die Gruppe *cmot* noch nicht definiert worden ist, ist sie separat dargestellt worden.

Die Unterteilung der Objekte in funktionalen Gruppen dient einerseits der Bezeichnung von Objekten und andererseits der Kennzeichnung der zu implementierenden Objekte.

Hierdurch werden Informationen darüber geliefert, welche Objekte in einem verwalteten Knoten zu implementieren sind, wenn die Funktion einer Gruppe für den verwalteten Knoten erforderlich ist. So brauchen z.B. Gateways keine Objekte zu unterstützen, die eine Endbenutzeranwendung betreffen, wie z.B. die elektronische Post.

Im folgenden werden die oben genannten funktionalen Gruppen kurz erläutert. Die Implementation von Gruppen ist in allen verwalteten Knoten als obligatorisch zu betrachten, falls keine andere Bedingung im Text angegeben ist. Die genauen Definitionen von Objekten sind in [McRo90, Rose90c, Rose91d] zu finden. Außerdem ist eine vollständige Liste der in der MIB-II definierten Objekte in „Anhang B. Liste der in MIB-II definierten Objekte" auf Seite 151 angegeben.

*Systemgruppe:*

```
system OBJECT IDENTIFIER ::= { mib 1 }
```

Die Informationen über das System werden mittels Objekten der Systemgruppe, die unter MIB-II durch vier Objekte ergänzt worden ist, aufbewahrt. Die Objekte der Systemgruppe sind:

- *sysDesc*: Beschreibung des Geräts,
- *sysObjektID*: Bezeichnung der Agenten-Software [Rose91a],
- *sysUpTime*: vergangene Zeit seit der letzten Initialisierung,
- *sysContact*: Name der Kontaktperson,
- *sysName*: Name des Geräts,
- *sysLocation*: Lage des Geräts und
- *sysServices*: ein Wert zur Bezeichnung der durch das Gerät zur Verfügung gestellten Dienste. Der Algorithmus zur Bestimmung dieses Werts ist in [Rose90c] angegeben.

*Schnittstellengruppe:*

```
interfaces OBJECT IDENTIFIER ::= { mib 2 }
```

Die Schnittstellengruppe umfaßt Informationen, die die Schnittstellen *(interfaces)* im System betreffen. Dabei wird das nicht-spaltenförmige[28] Objekt *ifNumber*, das die Anzahl der Schnittstellen des Knotens enthält, und die Tabelle *ifTable*, die aus spaltenförmigen Objekten besteht, für weitere Informationen über den verwalteten Knoten definiert. Objekte jeder Reihe der Tabelle sind beispielsweise:

- *ifDesc*: Beschreibung der Schnittstelle,
- *ifType*: Schnittstellentyp, wie z.B Ethernet oder FDDI,
- *ifMtu*: maximale Paketgröße,
- *ifSpeed*: Übertragungsrate in Bits pro Sekunde und
- *ifOperStatus*: gegenwärtiger Betriebszustand.

---

28 Unter einem nicht-spaltenförmigen Objekt versteht man ein Objekt, das nur eine Instanz besitzt, wie z.B. *sysDesc*. Entsprechend soll unter einem spaltenförmigen Objekt ein Objekt verstanden werden, das aus vielen Instanzen besteht, wie z.B. *ifPhysAddress*.

*Adressenumwandlungsgruppe:*

```
at OBJECT IDENTIFIER ::= { mib 3 }
```

Die Adressenumwandlungsgruppe, die auch AT[29]-Gruppe genannt wird, beinhaltet die Informationen für die Abbildung der Netzwerkadresse, wie z.B. IP-Adressen, auf die physikalische Adresse, wie z.B. MAC-Adressen. Diese Gruppe wurde in der MIB-II als "mißbilligt" bezeichnet und wird in der Zukunft nicht mehr existieren. Die Funktion der *atTable* wird dann durch die in MIB-II definierte neue Tabelle *ipAddrTable* erbracht. Der Grund für die Ansiedelung der Adreßinformationen in der IP-Gruppe ist die Protokollabhängigkeit der Netzwerkadresse [WiCa90]. Die drei spaltenförmigen Objekte dieser Gruppe sind:

- *atIfIndex*: Schnittstellennummer,
- *atPhysAddress*: physikalische Adresse der Abbildung und
- *atNetAddress*: Netzwerkadresse der Abbildung.

*IP-Gruppe:*

```
ip OBJECT IDENTIFIER ::= { mib 4 }
```

Die funktionale Gruppe IP *(Internet Protocol)* enthält Informationen über die IP-Schicht und setzt sich aus mehreren nicht-spaltenförmigen Objekte und drei Tabellen zusammen. Beispiele für nicht spaltenförmige Objekte sind:

- *ipForwarding*: Wirken als ein Gateway oder ein Host,
- *ipForwDatagrams*: Anzahl der weitergeleiteten Datagramme,
- *ipInReceives*: Anzahl der empfangenen Datagramme und
- *ipOutRequests*: Anzahl der gesendeten Datagramme.

Die drei in IP-Gruppe enthaltenen Tabellen heißen:

1. *ipAddrTable*: enthält die den verwalteten Knoten betreffenden IP-Adressen,
2. *ipRoutingTable*: setzt sich aus den dem verwalteten Knoten bekannten Leitwegen *(routes)* zusammen und
3. *ipNetToMediaTable*: bildet die IP-Adressen auf datenträgerspezifischen Adressen ab.

*ICMP-Gruppe:*

```
icmp OBJECT IDENTIFIER ::= { mib 5 }
```

Die funktionale Gruppe ICMP[30] *(Internet Control Message Protocol)* umfaßt Statistiken über die ICMP-Eingabe und -Ausgabe, und spezifiziert daher 26 Objekte vom Typ

---

29 *Address Translation*

30 ICMP ist ein Protokoll der IP-Schicht und wird zur Behandlung von Fehlern *(errors)* und Steuerungsnachrichten *(control messages)* benutzt [Jaco91].

*Counter*. Für jeden ICMP-Nachrichtentyp existieren zwei Objekte, die jeweils die Häufigkeit der Erzeugung einer Nachricht durch die IP-Instanz und des Empfangens derselben Nachricht von der IP-Instanz zählen, wie z.B.

- *icmpInDestUnreachs*: Anzahl der empfangenen ICMP-Nachricht, 'Ziel unerreichbar' *(Destination Unreachable)* und
- *icmpOutDestUnreachs*: Anzahl der gesendeten ICMP-Nachricht, 'Ziel unerreichbar' *(Destination Unreachable)*.

Außerdem existieren vier Objekte, die die Gesamtanzahl der geschickten, empfangenen, fehlerhaft empfangenen oder wegen Fehler nicht-geschickten ICMP-Nachrichten registrieren.

*TCP-Gruppe:*

```
tcp OBJECT IDENTIFIER ::= { mib 6 }
```

Die funktionale Gruppe TCP *(Transmission control protocol)* beinhaltet Informationen über die TCP-Schicht und ist aus mehreren nicht-spaltenförmigen Objekten und einer Tabelle aufgebaut. Beispiele für die nicht-spaltenförmigen Objekte sind:

- *tcpMaxConn*: maximale Anzahl der unterstützten TCP-Verbindungen,
- *tcpCurrEstab*: gegenwärtige Anzahl der bestehenden Verbindungen und
- *tcpInSegs*: Anzahl der empfangenen Segmente.

Die einzige Tabelle der TCP-Gruppe enthält Informationen über die TCP-benutzenden Anwendungsinstanzen. Schließlich muß die TCP-Gruppe nur in den verwalteten Knoten implementiert werden, die TCP implementiert haben.

*UDP-Gruppe:*

```
udp OBJECT IDENTIFIER ::= { mib 7 }
```

Die funktionale Gruppe UDP *(User Datagram Protocol)* enthält Informationen über die UDP-Schicht und besteht aus vier Objekten vom Type *Counter*:

- *udpInDatagrams*: Anzahl der dem Empfänger gelieferten Datagramme,
- *udpNoPorts*: Anzahl der empfangenen UDP-Datagramme, für die keine Anwendung am Bestimmungsort *(destination port)* vorhanden war,
- *udpInErrors*: Anzahl der nicht-zustellbaren UDP-Datagramme, die aus anderen Gründen als der fehlenden Anwendung *(udpNoPorts)* am Bestimmungsort empfangen worden sind und
- *udpOutDatagrams*: die Anzahl der ausgesendeten UDP-Datagramme.

Darüber hinaus besitzt die UDP-Gruppe eine Tabelle *udpTable*, die Informationen über die UDP benutzenden Anwendungsinstanzen umfaßt. Letztlich muß diese Gruppe nur

dann implementiert werden, wenn der verwaltete Knoten das Protokoll UDP implementiert hat.

*EGP-Gruppe:*

```
egp OBJECT IDENTIFIER ::= { mib 8 }
```

Die funktionale Gruppe EGP *(Exterior Gateway Protocol)* repräsentiert Informationen über die EGP-Partnerinstanzen *(peers)* und enthält vier Objekte vom Typ *Counter*, ein Objekt vom Typ *INEGER* und eine Tabelle:

Die Objekte vom Typ *Counter*:

- *egpInMsgs:*: Anzahl der fehlerfrei empfangenen EGP-Nachrichten,
- *egpInErrors*: Anzahl der fehlerhaft empfangenen EGP-Nachrichten,
- *egpOutMsgs*: Gesamtanzahl der lokal erzeugten EGP-Nachrichten und
- *egpOutErrors*: Gesamtanzahl der lokal erzeugten EGP-Nachrichten, die aufgrund der Ressourcenbeschränkungen in einer EGP-Instanz nicht gesendet worden sind.

Das Objekt vom Typ *INTEGER*:

- *egpAs*: autonome Systemnummer von der EGP-Instanz.

Die Tabelle *egpNeighTable* enthält Informationen über die EGP-Nachbarn dieser Instanz. Auch die EGP-Gruppe braucht nur dann implementiert zu sein, wenn im verwalteten Knoten das Protokoll EGP, das der Erreichbarkeit der autonomen Systeme dient, implementiert worden ist.

*Übertragungsgruppe*

```
transmission OBJECT IDENTIFIER ::= { mib 10 }
```

Die Übertragungsgruppe *(Transmission group)* gilt in der MIB-II als Platzhalter für die mediumspezifischen MIBs, wie z.B. die MIBs von IEEE 802.2, IEEE 802.5, FDDI und T1 zusätzlich zur Ethernet-Orientierung der MIB-I. Die mediumspezifischen MIBs werden zuerst der Experimentklasse zugeordnet und erst nach Annahme als Internet-Standard in der Übertragungsgruppe registriert. Abhängig von dem jeder Schnittstelle eines Systems unterliegenden Übertragungsmedium muß der entsprechende Teil der Übertragungsgruppe im System implementiert sein.

*SNMP-Gruppe*

```
snmp OBJECT IDENTIFIER ::= { mib 11 }
```

Die SNMP-Gruppe definiert neue Objekte, die statistische Informationen über die Netzwerkleistung für den Netzwerkverwalter beschaffen [Vand90]. Diese Informationen werden umso wichtiger, je mehr Größe und Komplexität des Netzwerks zunehmen. Die SNMP-Gruppe besteht aus einem Objekt vom Typ *INTEGER*, d.h.

- *snmpEnableAuthTraps*: zeigt an, ob der Agentenprozeß für die Erzeugung von Berechtigungsfehler-Traps *(authentication-failure traps)* konfiguriert ist,

und weiteren 29 Objekten vom Typ *Counter*, wie

- *snmpInPkts*: Anzahl der empfangenen SNMP-Pakete,
- *snmpInGetRequests*: Anzahl der verarbeiteten SNMP-Operation *get-request* oder
- *snmpInTotalSetVars*: Anzahl der geänderten MIB-Objekte.

Die Implementation der SNMP-Gruppe ist für alle Systeme obligatorisch, die eine SNMP-Protokollinstanz unterstützen.

#### 3.6.2.5 Erweiterungsmöglichkeiten

Die SMI wurde unter Berücksichtigung zweier Aspekte, der Einfachheit *(simplicity)* und der Erweiterbarkeit *(extensibility)* entwickelt. Die Einfachheit ist erzielt worden, indem einfache Datentypen, wie in „3.5 Datendarstellung" auf Seite 73 dargestellt wurde, verwendet worden sind. Eine Folge der Einfachheit ist, daß eine Erweiterung leichter zu realisieren ist.
Daneben ermöglicht die Erweiterbarkeit den zukünftigen Anforderungen gerecht zu werden und so eine größere Anzahl von Anwendungen und Technologien zu umfassen [Kral90]. Die SMI definiert drei Mechanismen zur Erweiterung der MIB:

1. Definition neuer Standard-Objekte innerhalb der Managementklasse *(mgmt)* durch Spezifizierung neuer Versionen der Internet-Standard-MIB, wie z.B. bei MIB-II als neue Version von MIB-I,

2. Definition von Experimentobjekten innerhalb der Experimentklasse *(Experimental)*, die weit verfügbar sind, aber keine Internet-Standard-Objekte repräsentieren, und

3. Definition von herstellerspezifischen Objekten innerhalb der Privatklasse *(private)* im Unternehmens-Unterbaum, wodurch die Hersteller ihre eigenen spezifischen Systeme in das Netzwerkmanagement integrieren können.

Die MIB-II als Beispiel für den ***ersten Mechanismus***, die sich zur Zeit in der Standardisierungsstufe *Proposed Standard* befindet und als Internet-Standard-MIB in Betracht gezogen wird, erweitert die Fähigkeiten des SNMP um verschiedene Datenübertragungstypen und Netzwerkgeräte. Die SNMP-Entwickler sind z.B. dabei, T1-Übertragungsgeräte, Token Ring, Token Bus, FDDI-Medium, Bridges und Terminals zu unterstützen. Schließlich wird die MIB in der Zukunft neue Objekte enthalten, die Anwendungen und Netzwerke gleich gut zu überwachen und zu steuern erlauben. Der Vorteil davon ist, daß die Benutzer ihre Anwendungen über das Netz überwachen und steuern können und in der Lage sein werden, dynamisch die Systemressourcen wie Rechenkapazität und Speicherbereich zu verwalten [Vand90].

Ein Ziel der MIB-II besteht z.B. in der Versicherung, daß der SNMP-Agent des einen Herstellers mit der SNMP-Management-Station, NMS, eines anderen Herstellers kommunizieren kann. Die MIB-II gibt also die Garantie, daß alle Hersteller das SNMP

für eine besondere Klasse von Geräten gleichartig implementieren, indem z.B. FDDI-Karten als Standard definiert werden. Als Ergebnis dürfte die MIB-II Inkompatibilitäten eliminieren helfen und den Nutzen des auf dem Standard basierenden Managements auf die heterogenen Netzwerke erweitern. Die drei in der SMI beschriebenen Regeln für die Definition von neuen MIB-Versionen sind:

1. neue Objekttypen dürfen unter neuen Namen definiert werden,

2. vorhandene Tabellen dürfen durch Zufügung einfacher Datentypen zu den schon in der Tabelle vorhandenen Listen ergänzt werden und

3. obwohl vorher definierte Objekttypen als "veraltet" *(obsolete)* gekennzeichnet werden, ist die Wiederverwendung ihrer Namen nicht erlaubt.

Während der Bedarf an herstellerspezifischen Erweiterungen immer vorhanden sein wird, wird Standardisierung der Schlüsseltechnologien die Arbeit des heterogenen Managements vereinfachen.

Durch den ***zweiten Mechanismus*** werden alle Objekte, die in Internet-Experimenten benutzt werden, definiert. So können Objekte durch die Internet-Gemeinschaft definiert, implementiert und getestet werden, ohne einer Standardisierungsprozedur unterworfen zu sein. Erst nachdem die Objekte sich in der Praxis als nützlich und funktional erwiesen haben, werden sie als Internet-Standard in Betracht gezogen.

Der ***dritte Mechanismus*** bietet den Herstellern die Möglichkeit, innerhalb der Privatklasse der MIB im Unternehmens-Teilbaum *(enterprises)* ihre SNMP-unterstützenden Geräte zu definieren, und diese Informationen mit den anderen Herstellern und Anwendern, die diese Informationen für Integration ihrer eigenen Systeme benötigen könnten, zu teilen.
Die Organisation der durch Hersteller definierten Objekte geschieht in der Art, daß jedem Hersteller, der sich beim *Information Sciences Institute, Marina Del Rey, Calif.* registriert hat, ein Teilbereich innerhalb des Unternehmens-Teilbaums zugewiesen wird. So wird dem Hersteller eine Unternehmensnummer zugeordnet, die seine Lage im Unternehmens-Teilbaum der Privatklasse festlegt. Der Vorteil dieses Verfahrens ist es, daß die Hersteller ihre Management-Anwendungen mit produktspezifischen MIB-Objekten den Wünschen der Kunden anpassen können. Dies gibt den Anwendern einen Überblick, wie z.B. ein Konzentrator konfiguriert ist oder welche Geräte installiert sind und wie sie funktionieren.

Zusammen mit diesem Trend in Richtung Anpassungsfähigkeit und Erweiterbarkeit der MIB in den Management-Agenten soll eine entsprechende Anpassungsfähigkeit in der NMS vorgesehen werden. Somit werden NMS-Anwender in der Lage sein, ihre Management-Software zu wechseln, wenn MIB-Erweiterungen eingebaut werden. Der Anwender kann den vollen Nutzen der Erweiterungen nicht in Anspruch nehmen, wenn das NMS ein Gerät zu überwachen und zu steuern versucht, ohne die MIB-Erweiterungen dieses Gerätes zu verstehen und auszunutzen. Hierdurch werden die Anwender im besten Falle nur teilweise Funktionalität erreichen. Außerdem beeinträchtigt das Fehlen einiger wesentlichen Werkzeuge *(tools)* das angemessene Management des Netzwerks [Vand90, SaHu90].

### 3.6.3 Simple Network Management Protocol (SNMP)

Das SNMP *(Simple Network Management Protocol)*, das in [CFSD90a] definiert worden ist, stellt das auf der Anwendungsschicht des Internetprotokollsatzes angesiedelte Kommunikationsprotokoll für das SNMP-Netzwerkmanagement dar. Dabei repräsentiert SNMP ein einfaches Protokoll, mit dessen Hilfe die ein Netzwerkelement betreffenden Management-Informationen geprüft und geändert werden können. Das auf SNMP, SMI und MIB basierende Netzwerkmanagement hat das Ziel, Netzwerk-Managern einen zentralen Punkt zur Beobachtung, Kontrolle und Verwaltung ihrer Installationen zu geben [Kral90, FeRS90, Benn89].

Das SNMP-Konzept wurde nicht von einer großen Herstellerabteilung entworfen, sondern im November 1987 aus dem SGMP *(Simple Gateway Management Protocol)* von vier Personen entwickelt [Sabo91]:

- *Jeffrey Case*, Universität Tennessee, SNMP Research, Inc.,
- *Chuck Davin*, Massachusetts Institute of Technology, MIT,
- *Mark Fedor*, NYSERNet und
- *Martin Schoffstall*, NYSERNet.

Das SGMP, beschrieben in [DCFS87], war ein früher Versuch, das Netzwerkmanagement in die TCP/IP-Umgebung einzuführen. Obwohl das SGMP zuerst für die Überwachung von Routern entwickelt wurde, überwachte es bald Systeme, wie z.B. Hosts oder Terminalserver. Die Forderung, das SGMP für den zukünftigen Übergang zu OSI vorzubereiten, wie es in einer von IAB initiierten Sitzung beschlossen wurde [Cerf88], und die Notwendigkeit der Erweiterung der Management Fähigkeiten von nur Überwachung auf Steuerung, machte Änderungen erforderlich, die das SGMP zum SNMP umgewandelt haben. So wurde das SNMP entwickelt, das dem SGMP in der Architektur und der Philosophie des Designs ähnelt, sich aber in der Syntax unterscheidet, so daß die beiden Protokolle inkompatibel sind [Howe90, CFSD89].

Unter SNMP ist also im übertragenen Sinne eine Sprache zu verstehen, mit deren Hilfe die Agenten und NMS miteinander kommunizieren [Scot90b]. So überträgt SNMP Management-Informationen zwischen NMS und Agenten in verwalteten Knoten, wie z.B. Hosts, Terminalserver, Bridges, Routers Gateways und dem Netz angeschlossenen Drucker. SNMP wurde zuerst zum Management von einem Netz von Netzwerken *(Internet)*, die auf TCP/IP-Protokollfamilie basieren, entwickelt worden, unterstützt aber zunehmend Netz von Netzwerken, die andere Protokollfamilien implementieren. Im SNMP wird u.a. beschrieben, welche Operationen der NMS zur Durchführung einer bestimmten Aufgabe mittels Agenten zur Verfügung stehen, oder im Falle eines Fehlers, welche Fehlermeldungen ein Agent zur Benachrichtigung der NMS benutzt.

Im folgenden werden die wesentlichen Elemente des SNMP, d.h. die Operationen, der Aufbau einer Nachricht und der Protokolldateneinheit und die Fehlermeldungen, im SNMP Traps genannt, erläutert.

### 3.6.3.1 Operationen

Die Operationen im SNMP-Netzwerkmanagement umfassen das Abfragen von Werten, das Setzen von Werten und das Versenden von Alarmmeldungen. Dabei ist ein Agent in der Lage, nach Empfang einer Anfrage von der NMS die Werte der MIB-Objektinstanzen zu prüfen und zu ändern oder im Falle eines außergewöhnlichen Fehlers die NMS zu benachrichtigen. Die vier im SNMP definierten Operationen sind:

1. *get*: wird zur Gewinnung von direkt durch ihren Namen gekennzeichneten Management-Informationen benutzt,

2. *get-next*: wird zur Gewinnung von Management-Informationen benutzt, die indirekt über den Namen der davorliegenden Objektinstanz im MIB-Baum bezeichnet werden,

3. *set*: erlaubt die Management-Informationen in der MIB zu ändern und

4. *trap*: ermöglicht, daß ein Agent über außergewöhnliche Ereignisse der NMS berichtet.

Die Beispiele für *get* und *get-next* werden in „4.5.1 Datenzugriffsmethode“ auf Seite 114 angegeben, wo stufenweise die Verbesserung des Datenzugriffs von einfacher serieller Methode bis zum Parallelalgorithmus beschrieben wird.

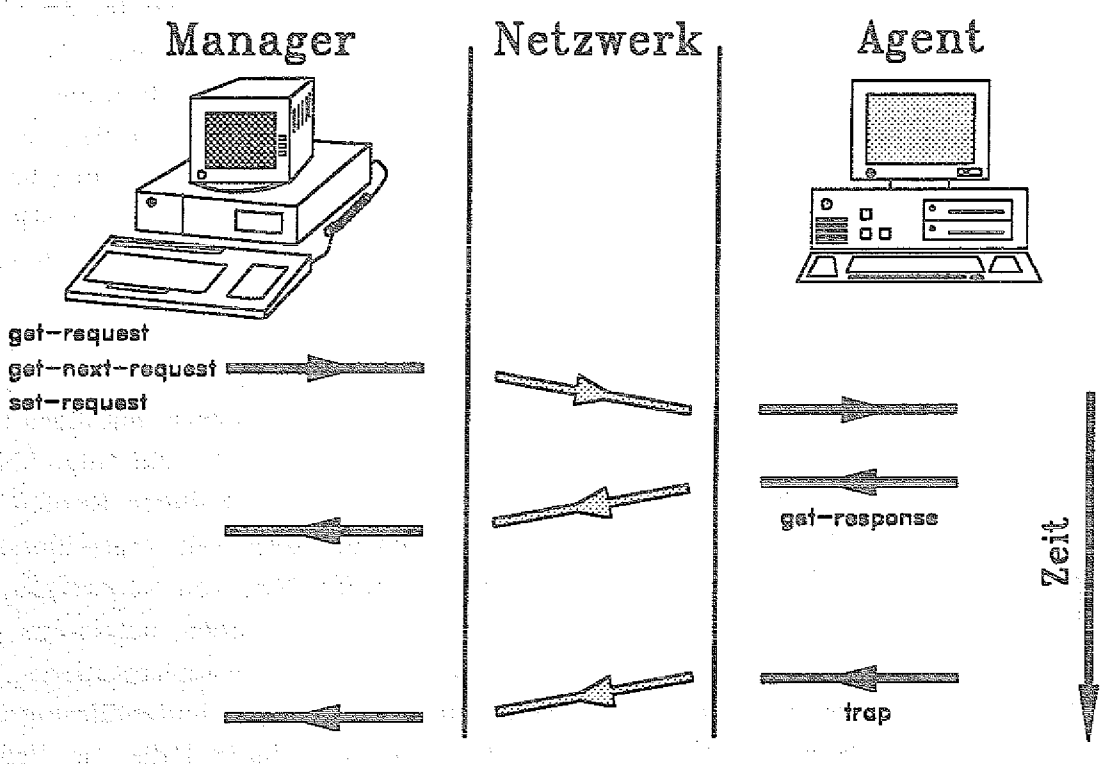



**Abb. 30. SNMP-Operationen:** Die Operationen in SNMP, welches ein asynchrones Anfrage/Antwort-Protokoll darstellt, werden durch Austausch der abgebildeten SNMP-PDUs realisiert.

Die Operation *set* ermöglicht die Definition von Imperativkommandos, die in der Regel eine komplexe Semantik besitzen, zu vermeiden. Die Imperativkommandos werden aber durch das Setzen einer Variable, die danach die gewünschte Aktion auslöst, realisiert. Ein Beispiel hierfür stellt das Kommando *reboot* dar, das im SNMP einfach durch das Setzen eines Variablenwertes verwirklicht wird, welcher die Zeit bis zum Neustarten des Systems angibt.
Das Setzen von Werten ist im SNMP als sicherheitskritisch anzusehen, da das SNMP nur begrenzt die Sicherheits- und Zugangsberechtigungsmechanismen unterstützt. Daher wird die Operation *set* durch einige Herstellerimplementierungen nicht unterstützt, um die Möglichkeit des Neustarten oder Abschalten eines Systems durch jeden Benutzer mit Zugang auf die NMS zu verhindern [Scot90b].

Die *Traps* werden im SNMP-Netzwerkmanagement hauptsächlich als ein Hilfsmittel zur Koordinierung des Zeitpunkts und zur Orientierung von Abfragen benutzt. Die Anzahl der Traps ist aus Gründen der Einfachheit auf sechs Arten begrenzt, die in „3.6.3.6 Traps" auf Seite 101 beschrieben werden.

### 3.6.3.2 Administrative Beziehungen

Die Zugriffsberechtigung zwischen den SNMP-Anwendungsinstanzen wird durch die in SNMP beschriebenen administrativen Beziehungen festgelegt. Dabei sind die SNMP-Anwendungsinstanzen solche Instanzen, die in der NMS oder in Netzwerkknoten angesiedelt sind und miteinander mittels SNMP kommunizieren.

| *Zugriffsmode* | *Zugriffsart entsprechend der Angabe in MIB* | | | |
|---|---|---|---|---|
| | Nur-Lesen | Lesen-Schreiben | Nur-Schreiben | unzugänglich |
| Nur-Lesen | C | C | A | A |
| Lesen-Schreiben | C | B | D | A |
| Anmerkung:<br>A: keine Operationen erlaubt<br>B: *get, get-next, set, trap*<br>C: *get, get-next, trap*<br>D: *get, get-next, set, trap* | | | | |

**Tab. 6. Gruppenprofil:** Die Klassen A, B, C und D stellen das Gruppenprofil, d.h. die erlaubten Operationen für ein Objekt dar. Werden in der Klasse D die Operationen *get* oder *trap* benutzt, so sind die empfangenen Werte der Objektinstanzen implementationsspezifisch [Rose91a].

Nach der Definition wird eine beliebige Anzahl von SNMP-Anwendungsinstanzen und ein Agent zu einer Gruppe *(community)* zusammengefaßt, die eine administrative Beziehung repräsentiert. Der Gruppe wird ein Name zugeordnet, der aus einer Folge von Oktetten besteht. Der Gruppenname *(community name)* ist in jeder SNMP-Nachricht enthalten und erlaubt auf dieser Weise, eine triviale Zugangsberechtigungsmethode zu realisieren. Die Methode besteht darin, daß geprüft wird, ob der Gruppenname in der Nachricht der Empfängerinstanz bekannt ist. Ist dies der Fall, so gilt die Senderinstanz als Mitglied der Gruppe. Nach der Erkennung der Mitgliedschaft der SNMP-Instanz in der Gruppe muß der verwaltete Knoten die erlaubte Zugriffsart bestimmen

[CFSD90a]. Die Zugriffsart auf jedes Objekt resultiert aus der Kombination der in der MIB definierten Zugriffsart und der in der Gruppe erlaubten Zugriffsmode, die in Tab. 6 dargestellt ist. Da in einer Gruppe nur die Zugriffsmoden 'Nur-Lesen' und 'Lesen-Schreiben' erlaubt sind, ergibt sich für jedes Objekt der Gruppe unter Berücksichtigung der möglichen in der MIB definierten Zugriffsarten ein Gruppenprofil *(community profile)*, das seinerseits die erlaubten Operationen für das Objekt spezifiziert.

Im Falle eines Vertretungsagenten wird eine Vertretungsgruppe *(proxy community)* alle das fremde Gerät betreffende Objekte umfassen. Die Vertretungsgruppe spezifiziert die Zugriffsmoden und definiert die erlaubten Operationen.

#### 3.6.3.3 Nachrichten

Das Protokoll SNMP regelt den Informationsaustausch zwischen den Agenten und der anfragenden NMS. Dabei werden die Management-Informationen in Datenpaketen, die SNMP-Nachrichten *(SNMP messages)* genannt werden, übertragen [Benn90].

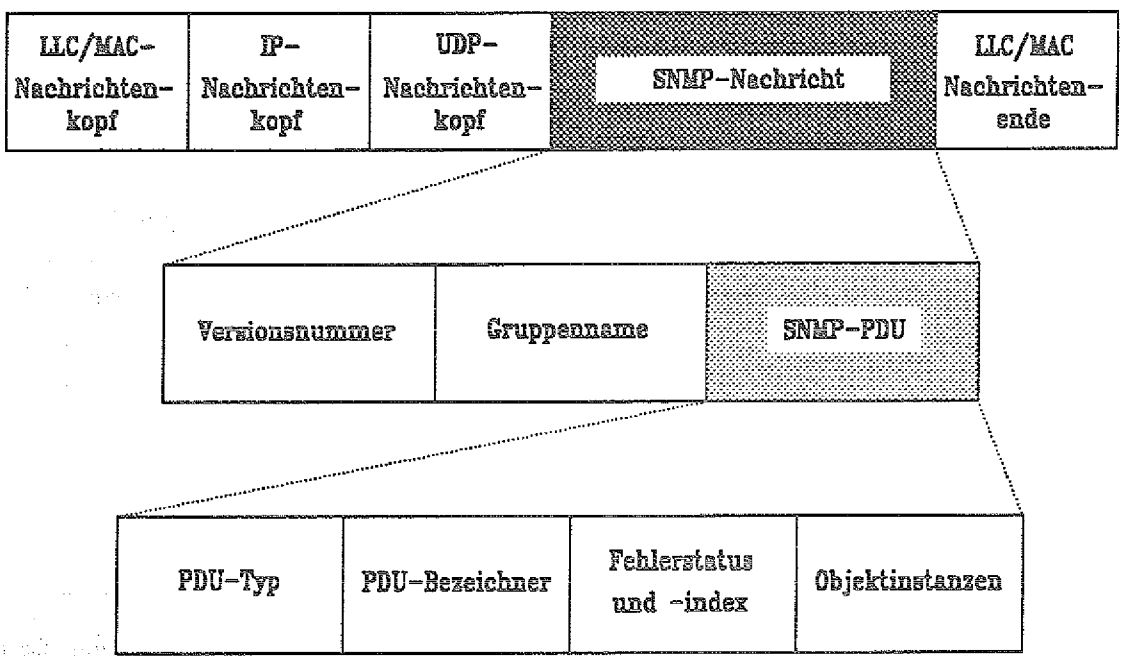



**Abb. 31. SNMP-Nachricht:** Abgebildet ist die in einem Netzwerkrahmen eingebettete SNMP-Nachricht und ihre Komponenten.

Eine SNMP-Nachricht ist aus drei Komponenten aufgebaut (vgl. Abb. 31):

1. Versionsnummer *(version number)*: dient zur Angabe der mit dem SNMP-Modul verbundenen Nummer, die gegenwärtig mit dem Wert eins belegt ist,

2. Gruppenname *(community string)*: ermöglicht die Überprüfung der Zugriffsberechtigung [Paul90] und

3. Daten: spezifiziert eine der fünf im SNMP definierten PDUs, die im nächsten Abschnitt beschrieben werden.

#### 3.6.3.4 Protokoll-Dateneinheiten

Im SNMP-Modul [CFSD90a] werden fünf Protokoll-Dateneinheiten *(Protocol Data Units, PDUs)* definiert:

1. *GetRequest-PDU*,
2. *GetNextRequest-PDU*,
3. *GetResponse-PDU*,
4. *SetRequest-PDU* und
5. *Trap-PDU*.

Dabei sind die vier ersten Protokoll-Dateneinheiten vom Datentyp PDU, während die *Trap-PDU* ihre eigene Syntax besitzt, die in „3.6.3.6 Traps" auf Seite 101 erläutert wird. Die Felder des Datentyps PDU sind:

- *request-id*: ist ein ganzzahliger Wert, der zur Kennzeichnung der Nachrichten verwendet wird,
- *error-status*: zeigt an, daß ein Fehler bei der Verarbeitung der Anfrage aufgetreten ist, falls das Feld nicht mit dem Wert *noError* belegt ist. Die anderen Werte sind:
  - *tooBig*: das Resultat einer Operation kann wegen der Größe nicht übertragen werden,
  - *noSuchName*: der mit der Operation übertragene Objektinstanzname ist unter Berücksichtigung des Gruppenprofils unbekannt,
  - *badValue*: der mit der Operation übergebene Objektinstanzwert ist inkorrekt,
  - *readOnly*: die Operation versucht den Wert einer Objektinstanz zu ändern, während der Wert nur gelesen werden darf,
  - *genErr*: andere, nicht erwähnte Fehler,
  - *error-index*: zeigt an, welche Objektinstanz in der Liste der Objekte *(variable-bindings)* den Fehler verursacht hat, falls der Wert nicht Null ist, und
  - *variable-bindings*: ist eine Liste von Objektinstanzen, wobei jede Objektinstanz durch ihren Namen und Wert dargestellt wird.

  Aufgrund der Semantik der Operationen *get* und *get-next* trägt der Wert einer Objektinstanz in den Datentypen *GetRequest-PDU* und *GetNextRequest-PDU* keine Information und wird von der Partnerprotokollinstanz ignoriert.

#### 3.6.3.5 Kommunikation zwischen Protokollinstanzen

Die Kommunikation zwischen Protokollinstanzen findet mittels Austausch von Nachrichten statt, wobei eine Nachricht völlig unabhängig von den anderen Nachrichten in-

nerhalb eines einzelnen UDP-Datagramms übertragen wird. Eine Protokollinstanz auf der Agentenseite empfängt alle Nachrichten über den UDP-Port 161 und schickt die Nachrichten, die eine Antwort enthalten, d.h. *GetResponse-PDU*, zu den Senderinstanzen und die Nachrichten, die Traps enthalten, zum UDP-Port 162 auf der Managerseite (vgl. Abb. 32) [CaMa91, Sabo91].

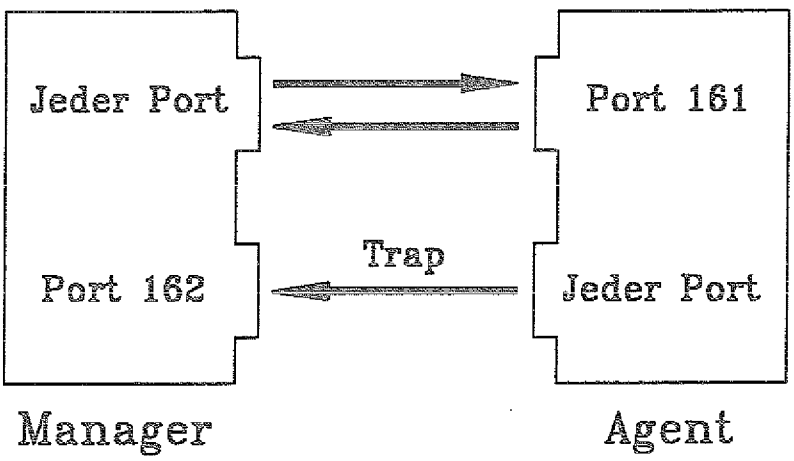



**Abb. 32. UDP-Portbelegung:** Die auf der Manager- und Agentenseite für den Empfang von SNMP-Nachrichten vorgesehenen Ports sind abgebildet. Alle Nachrichten werden auf der Agentenseite über den UDP-Port 161 empfangen. Die Traps werden aber nur über den UDP-Port 162 auf der Managerseite erwartet.

Die wesentlichen Phasen der Prozedur, die der Erzeugung einer Nachricht durch eine Protokollinstanz bis zum Empfang durch eine Protokollpartnerinstanz dienen, sind:

- Die Senderprotokollinstanz bildet die jeweilige Protokoll-Dateneinheit, die als ein ASN.1-Objekt zu betrachten ist, wie z.B. *GetRequest-PDU*;
- Das ASN.1-Objekt, Gruppenname und die Transportadresse des Senders und Empfängers werden dem Dienst übergeben, der das gewünschte Zugriffsberechtigungsschema implementiert. Die von diesem Dienst empfangene Antwort ist wieder ein ASN.1-Objekt;
- Die Protokollinstanz bildet eine Nachricht aus dem in voriger Stufe gewonnenen ASN.1-Objekt und dem Gruppennamen;
- Die Nachricht wird nach BER kodiert und mittels eines Transportdienstes, der im SNMP-Netzwerkmanagement durch das Protokoll UDP realisiert wird, zur Empfängerprotokollinstanz übertragen;
- Die Empfängerprotokollinstanz führt eine Syntaxprüfung des empfangenen Datagramms durch, bevor sie aus der empfangenen Nachricht das gesendete ASN.1-Objekt gewinnt. Ist die Syntaxprüfung erfolglos, so wird das Datagramm gelöscht und keine weitere Aktion unternommen;

- Die Versionsnummer der Nachricht wird geprüft und im Falle einer Unstimmigkeit das Datagramm gelöscht und keine weitere Aktion unternommen;

- Der Gruppenname, die Daten in der Nachricht und die Transportadresse des Senders und Empfängers werden dem Dienst übergeben, der das gewünschte Zugriffsberechtigungsschema implementiert. Als Resultat wird entweder ein ASN.1-Objekt oder ein Zugriffsberechtigungsfehler empfangen. Im letzteren Fall wird der Fehler von der Protokollinstanz registriert, möglicherweise ein Trap erzeugt und das Datagramm gelöscht;

- Das vom Zugangsberechtigungsdienst empfangene ASN.1-Objekt wird syntaktisch nachgeprüft, bevor daraus die jeweilige ASN.1-PDU gewonnen wird. Ein Fehler führt in diesem Fall dazu, daß das Datagramm gelöscht wird und keine weiteren Aktionen übernommen werden. Anderenfalls wird unter Benutzung des Gruppennamens das Gruppenprofil gewählt und die PDU entsprechend verarbeitet. Das Ergebnis der Verarbeitung wird, falls vorhanden, unter Zuhilfenahme der mit der Nachricht übertragenen Adressen der Senderprotokollinstanz zurückgeschickt.

#### 3.6.3.6 Traps

Das SNMP-Netzwerkmanagement überwacht das Netz durch trap-orientiertes Zyklisches Abfragen *(trap-oriented polling)*, das in „4.5.2 Zyklisches Abfragen vs. Berichten“ auf Seite 118 beschrieben wird. Im folgenden wird die Syntax von *Trap-PDU*, die im SNMP-Modul definiert ist, erläutert. Die *Trap-PDU* besteht aus den folgenden Feldern:

- *enterprise*: gibt durch das Objekt *sysObjektID* den Typ des trap-erzeugenden Objekts an,

- *agent-addr*: stellt mittels Datentyp *NetworkAddress* die Adresse des trap-erzeugenden Objekts dar,

- *generic-trap*: repräsentiert eine der folgenden in SNMP-Modul definierten Fehlermeldungen:

  - *coldStart*: zeigt an, daß die Senderprotokollinstanz dabei ist, sich neu zu initialisieren, wobei Änderungen bezüglich der Konfiguration des Agenten oder Protokollinstanzimplementation möglich sind,

  - *warmStart*: zeigt an, daß die Senderprotokollinstanz dabei ist, sich neu zu initialisieren, aber Änderungen im Bezug auf Konfiguration des Agenten oder Protokollinstanzimplementation nicht auftreten,

  - *linkDown*: gibt an, daß eine Schnittstelle ausgefallen ist, d.h. ihren Zustand von *up* zu *down* geändert hat,

  - *linkUp*: gibt an, daß eine Schnittstelle wieder in Funktion ist, d.h. den Zustand *up* besitzt,

- *authenticationFailure*: tritt auf, wenn eine SNMP-Nachricht von einer SNMP-Protokollinstanz empfangen wird, die nicht der Gruppe angehört,
- *epgNeighborLoss*: deutet darauf an, daß eine EGP-Partnerinstanz ausgefallen ist, d.h. im Zustand *down* ist, und
- *enterpriseSpecific*: spezifiziert die anderen außergewöhnlichen Ereignisse, die im Feld *specific-trap* gekennzeichnet werden.

- *specific-trap*: kennzeichnet, wenn es nicht den Wert Null besitzt, das unternehmensspezifische Trap *enterpriseSpecific*, das aufgetreten ist,
- *time-stamp*: gibt mittels des Objekts *sysUpTime* an, wann das Ereignis aufgetreten ist, und
- *variable-bindings*: ist eine Liste von Objektinstanzen, die Informationen über den Trap enthalten.

## 3.7 Transportmechanismus

Die durch SNMP-Protokollinstanz erzeugten SNMP-Nachrichten werden durch einen unzuverlässigen Datagrammdienst übertragen[31]. Obwohl das SNMP so entwickelt worden ist, daß es von dem unterliegenden Transportprotokoll unabhängig ist, entschied man sich zur Minimierung der Komplexität von Agenten für das UDP-Protokoll [CFSD90b].

Eine SNMP-Nachricht wird zur Übertragung zuerst in eine Folge von Oktetten kodiert *(serialization)*, die dann in das passende Feld des UDP-Datagramms eingebettet werden (vgl. Abb. 33). Auf der Empfängerseite werden die Oktette dekodiert, so daß die ursprüngliche Datenstruktur wieder gewonnen wird.

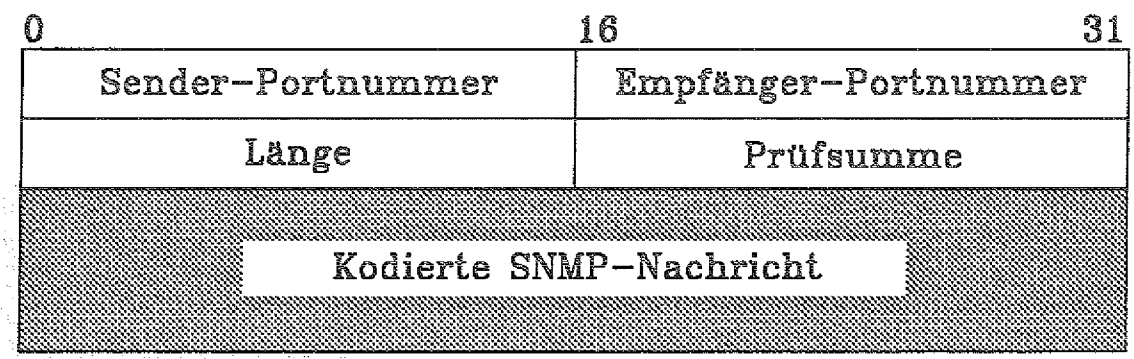



**Abb. 33. UDP-Datagramm:** Eine SNMP-Nachricht wird durch das verbindungslose Transport-Protokoll UDP übertragen. Hierfür wird das UDP-Datagramm benutzt, welches außer Sender- und Empfängeradressen und Kontrollinformationen, d.h. Länge und Prüfsumme, die kodierte SNMP-Nachricht enthält.

31 siehe auch „4.5.5 Transportprotokoll" auf Seite 129

Es ist zu berücksichtigen, daß eine SNMP-Nachricht unabhängig von anderen Nachrichten durch ein einzelnes UDP-Datagramm übertragen wird, und die SNMP-Implementierung nur die Nachrichten zuläßt, die nach der Kodierung eine Länge von maximal 484 Oktetten aufweisen.

# 4 SNMP-Netzwerkmanagement vs. OSI-Netzwerkmanagement

Durch die Einführung der offenen Systeme in den Netzwerkbetrieb ist die Notwendigkeit des Managements solcher Netzwerke, die diese Systeme benutzen, ständig gewachsen. Die Internet- und OSI-Protokolle sind für offene Systeme entwickelt worden und haben das Ziel, eine Netzwerktechnologie zu realisieren, die von einem speziellen Hersteller unabhängig ist, und die Interoperabilität bezüglich Produkten und Herstellerlösungen ermöglicht. Das Ziel der Standardisierungsorganisationen ist die Entwicklung eines Netzwerkmanagements, das herstellerübergreifend die komplexen aus offenen Systemen zusammengesetzten Netzwerke verwalten kann. Die zwei Netzwerkmanagement-Philosophien, nämlich das SNMP-Netzwerkmanagement und das OSI-Netzwerkmanagement, mit ihren Netzwerkmanagement-Protokollen, jeweils SNMP und CMIP genannt, stellen die Lösungen für das Management heterogener Netzwerke dar, und wurden bereits im zweiten bzw. dritten Kapitel beschrieben.

Dieses Kapitel hat die Aufgabe, die beiden Netzwerkmanagement-Philosophien und -Protokolle gegenüberzustellen. Dabei werden ihre Vorteile, Nachteile, Verbesserungen und Kompromißlösungen untersucht.

## 4.1 Vergleich der Architekturen

Eine wichtige Grundlage für den Vergleich der SNMP- und OSI-Netzwerkmanagement-Architekturen ist die Untersuchung ihrer Herkunft und Entwicklungsziele.
SNMP, das Netzwerkmanagement-Protokoll für die auf TCP/IP-basierenden Netzwerke, ist durch IAB *(Internet Activities Board)* als eine kurzfristige Lösung für das Management heterogener Netze entwickelt worden, wobei eine Migration zum OSI-Netzwerkmanagement-Rahmenwerk als langfristige Lösung beschlossen wurde [Cerf88]. Andererseits ist CMIP, welches das Netzwerkmanagement-Protokoll für die auf OSI-basierenden Netzwerke ist, durch ISO/IEC *(International Organization for Standardization / International Electrotechnical Commission)* als das Netzwerkmanagement-Protokoll der Zukunft entwickelt worden.

Die objektiven Kriterien, die eine gute Architektur von einer schlechten unterscheiden, sind schwierig zu erfassen, da die Industrie in hohem Maße politisch und marktorientiert ist. Es existieren aber Eigenschaften, die eine Netzwerkmanagement-Architektur grundsätzlich besitzen muß, wenn sie den Anforderungen der jetzigen und zukünftigen Netzwerk-Manager gerecht werden soll. Ein Beispiel hierfür ist ein Netzwerk mit vielen Knoten, die durch ein Netzwerkmanagement verwaltet werden sollen, wobei unter einem Knoten ein Supercomputer, eine Workstation, ein Terminal-Server, ein Router, ein Bridge oder ein Modem usw. verstanden werden kann. Ein nützliches Netzwerkmanagement muß also allgegenwärtig *(ubiquitous)* sein, d.h. es muß jeden Knoten verwalten können. Die schwierige Aufgabe des Netzwerkmanagements bei der Erfüllung der Allgegenwärtigkeitsbedingung wird deutlicher, wenn man die mögliche Existenz von unterschiedlichen Knoten in einem Netzwerk betrachtet.

Ein weiterer wichtiger Punkt ist, daß ein Knoten nicht durch das Management in seiner Funktionalität beeinflußt werden soll. So wird z.B. ein Gateway für seine Funktionalität als Gateway gekauft und betrieben und nicht, um das Netzwerkmanagement-Protokoll

darauf laufen zu lassen. Unter Berücksichtigung der Allgegenwärtigkeitsbedingung muß die Netzwerkmanagement-Architektur so entworfen sein, daß die entstandene Technologie auf einen Knoten mit dem niedrigsten Niveau an Fähigkeiten angewandt werden kann, ohne seine Leistung zu verschlechtern. Die obigen Überlegungen werden in einem fundamentalen Axiom *(fundamental axiom)* festgehalten[32]:

> Die Wirkung eines Netzwerkmanagements auf einen verwalteten Knoten muß minimal sein, was den kleinsten gemeinsamen Nenner widerspiegelt.

Eine Netzwerkmanagement-Architektur soll also eine Technologie beschreiben, die auf allen Knoten implementierbar und effektiv ausgeführt werden kann [Rose91b].
Jede Architektur bestimmt, in welchem Maße die Funktionalität zwischen Management-Stationen und verwalteten Knoten verteilt werden soll. Eine gute Architektur realisiert die Technologie, die die Anforderungen an verwaltete Knoten vermindert, und sie zu den Management-Stationen hin verlagert. Die Verschiebung der Funktionalität zu den Management-Stationen sorgt für die Wirtschaftlichkeit der Technologie, da die Zahl der verwalteten Knoten im Vergleich zu der der Management-Stationen überwiegt.

Die SNMP-Netzwerkmanagement-Architektur ist einfach und fähig, die grundlegenden Aufgaben des Netzwerkmanagements zu erfüllen. Diese Merkmale sorgten für eine schnell und leicht implementierbare Technologie und dadurch für eine rasche Verbreitung, was den Erfolg dieser Architektur kennzeichnet. Im Gegensatz dazu hat die aus der OSI-Netzwerkmanagement-Architektur resultierende komplexe Technologie bewirkt, daß diese nur schwer zu verstehen und zu implementieren ist. Dadurch ist die Verbreitung der OSI-Netzwerkmanagement-Technologie verlangsamt worden, wobei auch die Seltenheit der auf den OSI-Protokollturm basierenden Netzwerke eine wichtige Rolle gespielt hat. Die durch die Implementation der OSI-Netzwerkmanagement-Technologie an einfache Netzwerkgeräte gestellten Forderungen stehen im Gegensatz zum fundamentalen Axiom einer übergreifenden Netzwerkmanagement-Architektur und sind als Nachteil anzusehen.

## 4.2 Vergleich der Rahmenwerke

Die Standards, die das SNMP- und das OSI-Netzwerkmanagement-Rahmenwerk spezifizieren, lassen die strukturellen Ähnlichkeiten zwischen den beiden Netzwerkmanagement-Rahmenwerken erkennen. Das SNMP-Netzwerkmanagement-Rahmenwerk wird durch die drei folgenden Dokumente charakterisiert:

1. Standard für die Definition der Struktur von Management-Informationen *(Structure of Management Information, SMI, RFC1155)* [RoMc90a],
2. Standard für die Definition von verwalteten Objekten *(Management Information Base, MIB, RFC1156)* [McRo90] und
3. Standard für die Kommunikation *(Simple Network Management Protocol, SNMP, RFC1157)* [CFSD90a].

---

32 siehe Seite 71 in [Rose91a]

Das OSI-Netzwerkmanagement-Rahmenwerk basiert im wesentlichen auf folgenden Dokumenten:

Standards für die Architektur

- Management-Rahmenwerk *(Management Framework)* [ISO7498-4],
- System-Management-Überblick *(Systems Management Overview)* [ISO10040],

Standards für die Management-Information

- Management-Informationsmodell *(Management Information Model)* [ISO10165-1],
- Definition von Management-Information *(Definition of Management Information)* [ISO10165-2],
- Richtlinien für die Definition von verwalteten Objekten *(Guidelines for the Definition of Managed Objects)* [ISO10165-4],

Standards für die Kommunikation

- Allgemeiner Management-Informationsdienst *(Common Management Information Service, CMIS)* [ISO9595],
- Allgemeines Management-Informationsprotokoll *(Common Management Information Protocol, CMIP)* [ISO9596],

Standards für die System-Management-Funktionen

- Objekt-Management-Funktion *(Object Management Function)* [ISO10164-1],
- Zustand-Management-Funktion *(State Management Function)* [ISO10164-2],
- Attribute für Darstellung von Beziehungen *(Attributes for Representing Relationships)* [ISO10164-3],
- Alarmmeldungs-Funktion *(Alarm Reporting Function)* [ISO10164-4],
- Ereignisbericht-Management-Funktion *(Event Report Management Function)* [ISO10164-5],
- Log-Steuerungs-Funktion *(Log Control Function)* [ISO10164-6],
- Sicherheits-Alarmmeldungs-Funktion *(Security Alarm Reporting Function)* [ISO10164-7],
- Sicherheitsprüfpfad-Funktion *(Security Audit Trail Function)* [ISO10164-8],
- Objekte und Attribute für Zugriffssteuerung *(Objects and Attributes for Access Control)* [ISO10164-9],
- Abrechnungszähler-Funktion *(Accounting Meter Function)* [ISO10164-10],
- Kapazitätüberwachungs-Funktion *(Workload Monitoring Function)* [ISO10164-11],
- Test-Management-Funktion *(Test Management Function)* [ISO10164-12],
- Meßzusammenfassungs-Funktion *(Measurement Summarisation Function)* [ISO10164-13],

Standards für die Definition von verwalteten Objekten

- IS-IS Wegewahl-Austausch-Protokoll[33] *(IS-IS Routing Exchange Protocol)* (ISO 10589),
- Elemente von Management-Information verbunden mit den OSI-Vermittlungsschicht-Standards *(Elements of Management Information related to OSI Network Layer Standards)* (ISO 10733) und
- Elemente von Management-Information verbunden mit den OSI-Transportschicht-Standards *(Elements of Management Information related to OSI Transport Layer Standards)* (ISO 10737).

Dem Rahmenwerk einer Netzwerkmanagement-Architektur ist eine sehr wichtige Rolle zuzuordnen, da es die Schnelligkeit der Entwicklung, Standardisierung, Implementation und schließlich Fertigstellung von Produkten beeinflußt. Es wird sogar behauptet, daß mit einem richtigen Rahmenwerk das Netzwerkmanagementproblem gelöst sei [Rose91b].
Der Übergang von SNMP- zum OSI-Netzwerkmanagement-Rahmenwerk ist ähnlich der geplanten Fortentwicklung von TCP/IP- nach OSI-Protokolltürmen vorgesehen. Der Nachteil dieser Lösung ist, daß ein Übergang von der kurzfristigen Lösung zur langfristigen Lösung erforderlich ist [Rose91a]. Deshalb mußten die Rahmenwerke so entworfen werden, daß der Übergang der Technologien in Zukunft gut möglich ist. Um dieses Ziel zu erreichen, hat man sich z.B. für eine gemeinsame Sprache, nämlich ASN.1, entschieden, die der Definition von verwalteten Objekten dient und daher die Übertragung der Management-Informationen unabhängig vom benutzten Protokoll ermöglicht.

Das SNMP-Netzwerkmanagement-Rahmenwerk sollte den Anwendern möglichst schnell ein Werkzeug zur Erfüllung der real existierenden Management-Aufgaben zur Verfügung stellen. Deshalb wurde problemorientiert in einem gut definierten Problembereich gearbeitet, und dabei kein großer Wert auf die Globalität der Technologie im Vergleich zur OSI-Netzwerkmanagement-Technologie gelegt. Aufgrund der Einfachheit *(simplicity)* des Rahmenwerks, der resultierenden Technologie und der endlichen Verwendbarkeit *(scalability in deployment)* wurden in einem begrenzten Zeitrahmen die entstandenen Lösungen in verschiedenen Umgebungen implementiert und Verbesserungen vorgenommen. Erst nach der praktischen Überprüfung wurden sie dem Standardisierungsprozeß unterworfen und können deshalb als De-facto-Standards bezeichnet werden.

Das OSI-Netzwerkmanagement-Rahmenwerk wurde im Hinblick darauf entwickelt, daß das OSI-Netzwerkmanagement eine langfristige Lösung sein soll. Das Ziel der Entwicklung ist die Schaffung einer relativ vollständigen Lösung für das Netzwerkmanagementproblem in heterogenen Umgebungen, d.h. eine Lösung, die im Gegensatz zum SNMP-Netzwerkmanagement alle möglichen Forderungen an das Netzwerkmanagement umfassen soll. Die Folge ist, daß die Standardisierungsarbeit bei der Entwicklung und Verabschiedung von Standards langsam und schwierig wird, da viele nationale und internationale Organisationen zusammenarbeiten und sich einigen müssen.

---

33 Ein Zwischensystem *(Intermediate System, IS)* tauscht mit einem anderen Zwischensystem mit Hilfe dieses Protokolls Wegewahl-Informationen aus. Unter einem Zwischensystem versteht man ein OSI-System, das kein Endsystem ist und der Kommunikation zwischen den Endsystemen dient, wie Repeater, Bridge und Router [Jaco91].

Im Unterschied zu den SNMP-Netzwerkmanagement-Standards werden die OSI-Netzwerkmanagement-Standards zuerst gebilligt und dann in der Praxis getestet und implementiert. Sie können deshalb De-jure-Standards genannt werden.

Ein Ergebnis der oben dargestellten unterschiedlichen Annäherung zum Netzwerkmanagementproblem ist, daß das SNMP-Netzwerkmanagement-Rahmenwerk eine umgängliche Technologie *(tractable technologie)* hervorgebracht hat, während die aus dem OSI-Netzwerkmanagement-Rahmenwerk entstandene Technologie, soweit sie vorhanden und implementiert worden ist, weniger leicht handhabbar ist.
Die Problematik mit dem OSI-Netzwerkmanagement-Rahmenwerk ist unter anderem die Inanspruchnahme eines großen Zeitrahmens für die Entwicklung der Technologie, die für einen Verlust an Wettbewerbsfähigkeit im Markt gesorgt hat. Diese komplexe Technologie erfordert ihrerseits von den verwalteten Knoten relativ hohe Rechen- und Speicherkapazitäten, die nicht jeder verwaltete Knoten erbringen kann.
Nur die Technologien, die innerhalb eines gewissen Zeitrahmens implementiert werden, besitzen die Möglichkeit, in verschiedenen Umgebungen eingesetzt und dadurch nach Fehlerbehebung verbessert und robuster zu werden, da der Wettbewerb von robusten Produkten die Existenz eines reifen Marktes kennzeichnet [Rose91b].

## 4.3 Vergleich der SMIs

Während die Probleme der Speicherkapazität und der Bandbreite im Zuge der neuen, fortschrittlichen und preisgünstigen Technologien ihre entscheidende Rolle als Barrieren für Netzwerkmanagement verlieren, nimmt die Komplexität von Netzwerken und damit die Schwierigkeiten des Managements dieser Netzwerke ständig zu. Um die komplexen, heutigen und zukünftigen Netzwerke verwalten zu können, entschied man sich für eine objektorientierte Vorgehensweise, die es erlaubt, ein Netzwerk Stück für Stück, anstatt in seiner ganzen Komplexität, zu betrachten [CrWa91].

Ein wichtiger Bestandteil der SNMP- und OSI-Netzwerkmanagement-Rahmenwerke ist die Spezifizierung der Struktur der Management-Information *(Structure of Management Information, SMI)*.
Das SNMP- und das OSI-Netzwerkmanagement-Rahmenwerk benutzen beide für die Beschreibung von Management-Informationen die objektorientierte Sprache ASN.1. Der Unterschied liegt nur darin, daß das SNMP-Netzwerkmanagement-Rahmenwerk lediglich eine Teilmenge dieser Sprache verwendet [CFSD90b]. Daher liefert CMIP den Management-Anwendungen eine leistungsfähige, objektorientierte Schnittstelle, während SNMP eine einfache und mehr attributorientierte Schnittstelle zur Verfügung stellt. In diesem Sinne ist also das SNMP-Netzwerkmanagement-Rahmenwerk nicht objektorientiert, sondern attributorientiert.
Durch die objektorientierte Betrachtung werden verschiedene Netzwerkgeräte, wie z.B. Modems, Bridges und Routers, als verwaltete Objekte aufgefaßt. Hierdurch wird eine gewisse Abstraktion erzeugt, die einerseits die Standards verallgemeinert und auf allen Umgebungen anwendbar macht, und andererseits wegen der Abstraktheit das Verständnis der OSI-Diskussion erschwert.

Die verwalteten Objekte im OSI-Netzwerkmanagement sind umfassender als die verwalteten Objekte im SNMP-Netzwerkmanagement. Die Objekte im OSI-Netzwerkma-

nagement besitzen Attribute *(attributes)*, melden Ereignisse *(events)* und führen festgelegte imperative Aktionen aus. Außerdem stehen die Objekttypen durch die Vererbungs-Beziehung *(inheritance relationship)* und die eigentlichen verwalteten Objektinstanzen durch die Enthaltungs-Beziehung *(containment relationship)* miteinander in Beziehung. Die Objekte im SNMP-Netzwerkmanagement sind den Attributen von Objekten im OSI-Netzwerkmanagement ähnlich und in Gruppen organisiert [Pres90].

Für die Benennung von verwalteten Objekten benutzt das OSI-Netzwerkmanagement das Enthaltungsmodell, d.h. die Objekte werden als ineinander enthalten betrachtet[34]. Im OSI-Netzwerkmanagement besteht die Möglichkeit, mehrere verwaltete Objektinstanzen vom gegebenen Typ zu erzeugen oder zu entfernen. Daher kann eine Anwendung die Objektinstanzen aus der Entfernung mittels sinnvoller Namen verwalten, die an keinen bestimmten Adressierungsmechanismus gebunden sind. Im SNMP-Netzwerkmanagement dagegen werden die verwalteten Geräte mittels IP-Adressen gekennzeichnet, und die verwalteten Objekte innerhalb dieser Geräte werden durch vorher definierte Namensbäume benannt [BeCW90].

Das OSI-Netzwerkmanagement-Rahmenwerk stellt das Dokument "Richtlinien für die Definition von verwalteten Objekten" *(Guidelines for the Definition of Managed Objects)* [ISO10165-4] zusammen mit ASN.1 als eine objektorientierte Sprache zur Verfügung, um die oben genannten Eigenschaften eines verwalteten Objekts definieren zu können. Das SNMP-Netzwerkmanagement-Rahmenwerk stellt zur Beschreibung der Struktur von Management-Informationen das Dokument "Struktur und Identifikation von Management-Information für die auf TCP/IP-basierenden Internets" *(Structure and Identification of Management Information for TCP/IP-based Internets, RFC1155)* [RoMc90a] zur Verfügung. Dieses Dokument beschreibt:

- die Identifizierung von verwalteten Objekten,
- die Beziehung zwischen den Namen von verwalteten Objekten für z.B. effektives Durchqueren der MIB,
- die Syntax von verwalteten Objekten, d.h. den das verwaltete Objekt modellierenden Datentyp,
- die Art des Zugriffs auf verwaltete Objekte, wie z.B. nur Lesen, Schreiben-Lesen und
- den Status, d.h. die Implementationsforderung für verwaltete Objekte, wie z.B. fakultativ *(optional)* oder obligatorisch *(mandatory)*.

Im Gegensatz zum OSI-Netzwerkmanagement wird im SNMP-Netzwerkmanagement nur eine Teilmenge der objektorientierten Sprache ASN.1 und der Kodierungssprache BER *(Basic Encoding Rule)* benutzt. Der Vorteil dieser Strategie besteht unter anderem darin, daß die Implementierung, durch kurzgefaßte Informationen, vereinfacht wird. Ein Beispiel hierfür stellt die vom SNMP-Netzwerkmanagement verwendete Teilmenge von

---

34 siehe „2.4.2.3 Verwaltetes Objekt" auf Seite 17

ASN.1-Datentypen dar, die im „3.5 Datendarstellung“ auf Seite 73 beschrieben wurden. Dabei werden z.B. die folgenden ASN.1-Datentypen vom SNMP-Netzwerkmanagement nicht benutzt:

```
BOOLEAN
BIT STRING
ObjectDescriptor
EXTERNAL
REAL
ENUMERATED
SET
SET OF
```

Durch die oben beschriebene Begrenzung der Datentypen im SNMP-Netzwerkmanagement auf einen Minimum müssen die nicht erlaubten Datentypen, wenn nötig, durch die erlaubten realisiert werden. Ein BOOLEAN kann z.B. durch einen INTEGER realisiert werden, der nur die Werte Null für falsch und Eins für wahr haben kann. Der Nachteil ist, daß ein Eintrag in der Adreßtabelle im SNMP-Netzwerkmanagement nicht als ein selbständiges Objekt sondern als eine Folge von Objekten definiert werden muß. Durch die Beschränkung der Datentypen im SNMP-Netzwerkmanagement kann der Aufwand bezüglich Programmentwicklung, Kodegröße und Ausführungszeit minimiert werden.

Die Objekte im SNMP-Netzwerkmanagement werden den vier Objektklassen Verzeichnis-, Management-, Experiment- und Privatklasse zugeordnet. So können die Objekte stufenweise erweitert werden. Die Experimentklasse erlaubt die gründliche Erprobung der Objekte, bevor sie Aufnahme in die vorgeschriebene Objektklasse finden. Im Gegensatz dazu sind die Objekte im OSI-Netzwerkmanagement keinen vorausgehenden Experimenten und Tests unterworfen.
Für die Definition von verwalteten Objekten wird in SNMP-SMI ein ASN.1-Makro, OBJECT-TYPE-Makro, verwendet, das aber nicht beschreibt, wie SNMP die Instanzbezeichner bildet [35]. Dieser Nachteil kann beseitigt werden, wenn die Makro-Definition um einen Eintrag INSTANCE, wie im folgenden beispielsweise für ifEntry-Objekt dargestellt ist, ergänzt wird[36]:

```
ifEntry SNMP-OBJECT-TYPE
        SYNTAX  IfEntry
        ACCESS  read-only
        STATUS  mandatory
        INSTANCE { ifIndex }
        ::= { ifTable 1 }
```

Während das objektorientierte Modell als die beste Wahl für die Beschreibung der Netzwerkmanagementprobleme erscheint, argumentieren die Befürworter des

---

35 siehe Seite 126 in [Rose90c]

36 siehe Seite 166 in [Rose91a]

SNMP-Netzwerkmanagements, daß die Wahl des objektorientierten Modells falsch gewesen ist[37]:

> Eine objektorientierte Annäherung könnte für die Programmiersprachen der vierten Generation schön sein, aber sie ist eine schlechte Grundlage, wenn dabei versucht wird, ein reales Netzwerk zu verwalten.

## 4.4 Vergleich der MIBs

Die Management-Informationsbasis *(Management Information Base, MIB)*, die eine Ansammlung der Definitionen von verwalteten Objekten darstellt, wird vom SNMP- und vom OSI-Netzwerkmanagement-Rahmenwerk unterschiedlich spezifiziert.

Das OSI-Netzwerkmanagement-Rahmenwerk betrachtet die MIB als einen Ort zur Aufbewahrung von Management-Informationen, die durch OSI-Management-Protokolle übertragen und beeinflußt werden dürfen. Während die abstrakte Syntax und Semantik der in einer MIB vorhandenen Informationen der Normung unterworfen sind, existiert keine Einschränkung bezüglich der physikalischen Speicherung der Management-Informationen.
Das OSI-Netzwerkmanagement-Rahmenwerk setzt keine Beschränkung bezüglich der Anzahl der zu definierenden verwalteten Objekte und trägt dadurch zur Kompliziertheit der Technologie bei. Während das OSI-Netzwerkmanagement-Protokoll, CMIP, die dynamische Erzeugung von MIBs erlaubt, wie z.B. Alarme, ist dies im SNMP nur durch Überlisten der Architektur möglich[38].

Das SNMP-Netzwerkmanagement-Rahmenwerk definiert eine bestimmte Anzahl von verwalteten Objekten, die als Kern der zu definierenden verwalteten Objekte bezeichnet werden dürfen und in acht Gruppen unterteilt sind. Der Grund für die Auswahl einer minimalen Anzahl von Objekten ist die Einhaltung der Einfachheit *(simplicity)* als ein charakteristisches Merkmal des SNMP-Netzwerkmanagement-Rahmenwerks gewesen. Hierdurch wird die wichtige Eigenschaft, nämlich die Erweiterbarkeit *(extensibility)* nicht verletzt, da der SMI-Standard (RFC1155, [RoMc90a]) drei Möglichkeiten zur Integration von neuen verwalteten Objekten definiert:

1. Definition einer neuen Version von MIB, die neu normierte verwaltete Objekte umfaßt, wie z.B. MIB-II (*RFC1158 veraltet, RFC1213*, [Rose90c, Rose91d]),

2. Definition von nicht normierten Objekten durch den Versuchs-Teilbaum *(experimental subtree)* und

3. Definition von privaten verwalteten Objekten durch den Unternehmens-Teilbaum *(enterprises subtree)*.

---

37 siehe Seite 9 in [Rose91b]

38 siehe Seite 80 in [Doug91]

Die MIB-Module, die nach den SMI-Regeln des SNMP-Netzwerkmanagement-Rahmenwerks aufgebaut sind, definieren verwaltete Objekte, die einen numerischen Bezeichner haben und in lexikographischer Ordnung aufgelistet sind. Die Module definieren keine fakultativen verwalteten Objekte, sondern verwaltete Objekte, die miteinander in Beziehung stehen und in Gruppen zusammengefaßt sind. Die Gruppen stellen eine Grundeinheit derart dar, daß die Implementation einer Gruppe in einem verwalteten Knoten die Implementation aller anderen Objekte dieser Gruppe in diesem Knoten erzwingt. Wenn z.B. die Gruppe EGP in einem Gateway implementiert wird, müssen auch alle übrigen Objekte dieser Gruppe in diesem Gateway implementiert sein.
Eine Definition von fakultativen verwalteten Objekten würde die Wahrscheinlichkeit der Unterstützung von Objekten durch Manager- und Agent-Implementationen bei einer großen Anzahl von Objekten erniedrigen, und dadurch die Interoperabilität von Systemen beeinträchtigen [Pres90].

Der Vorteil der Definition eines MIB-Modul-Kerns für die notwendigen Objekte ist, daß die wichtigsten Probleme durch konzentriertes Arbeiten zuerst gelöst werden und der Weg zur Implementation schneller zurückgelegt wird. Nach der Implementation eines Netzwerkmanagements lassen sich viel effektiver neue verwaltete Objekte definieren, da die Anforderungen an die Objekte relativ genau festliegen.

## 4.5 Vergleich der Protokolle und Mechanismen

Das SNMP- und das OSI-Netzwerkmanagement-Rahmenwerk definieren jeweils die Kommunikationsprotokolle SNMP und CMIP, die für das Management heterogener Netzwerke entworfen worden sind. Die Protokolle wurden in den jeweiligen Kapiteln einzeln beschrieben. Jetzt wird kurz auf die Ähnlichkeiten und Differenzen zwischen den beiden Protokollen eingegangen.

Die beiden Protokolle SNMP und CMIP sind sich in vieler Hinsicht ähnlich. Die wesentlichen Ähnlichkeiten können wie folgt zusammengefaßt werden [Pres90]:

- beide Protokolle dienen der Übertragung von Management-Informationen in einem Netzwerk, wobei die Überwachung und Steuerung von Netzwerkgeräten ermöglicht wird, d.h. „sie sind ähnlich, indem sie beide das gleiche Ziel haben“[39].

- Als Management-Protokolle für heterogene Netze sind sie beide für die Behandlung von Aufgaben, wie z.B. Problemdiagnose, Kapazitätplanung usw., nützlich.

- Die beiden Protokolle SNMP und CMIP benutzen das Konzept der MIB und stellen zusätzlich die Möglichkeit der herstellerspezifischen MIB-Erweiterungen zur Verfügung. Hierdurch kann eine große Anzahl von Netzwerkgeräten unterstützt werden, wobei das Management der heterogenen Netze ermöglicht wird. Schließlich existiert der Wunsch, eine Kompatibilität zwischen den SNMP-MIBs und den CMIP-MIBs zu erreichen, um die mögliche langfristige Migration zum OSI-Netzwerkmanagement zu erleichtern.

---

[39] siehe Seite 184 in [Fish91]

- Beide Management-Protokolle werden durch die Benutzung der OSI-ASN.1-Notation und der OSI-Grundkodierungsregeln *(Basic Encoding Rules, BER)* spezifiziert.

- Die Möglichkeit der Verwaltung eines Netzwerkgerätes, das SNMP oder CMIP nicht unterstützt, wird von den beiden Protokollen durch einen Vertretungsagent *(proxy agent)* zur Verfügung gestellt.

- Die beiden Protokolle haben die Operationen SET, GET und die Operation zur Ereignismeldung, die EVENT in CMIP und TRAP in SNMP heißt, gemeinsam.

Die Netzwerkmanagement-Protokolle SNMP und CMIP unterscheiden sich im wesentlichen durch

- die Art des Datenzugriffs,
- den Ereignisbenachrichtigungsmechanismus,
- den Umfang der Funktionalität,
- die beanspruchte Speichergröße und Leistung,
- den unterliegenden Transportmechanismus,
- die Standards und Tests und
- die Verfügbarkeit der Produkte [Bona90a].

Zur Verdeutlichung der Unterschiede zwischen den beiden Protokollen wird von Jeffrey D. Case, einem der SNMP-Autoren, behauptet[40]:

> Die Differenzen zwischen SNMP und CMIP sind von der Kategorie der Differenzen zwischen C und Ada.

### 4.5.1 Datenzugriffsmethode

Im SNMP stehen die zwei Operationen *get* und *get-next* zur Verfügung, die zur Informationsgewinnung, d.h. zum Zugriff auf die Management-Informationen in der MIB benutzt werden. Diese Operationen werden in erster Linie durch SNMP zur Gewinnung von einzelnen individuellen Informationen benutzt. Um die Informationsgewinnung selektiver zu gestalten, wurde ein Werkzeug mit dem Namen *SNMP Query Language* entwickelt [Carl91, Yeon90].
SNMP stellt keinen Mechanismus für seine Operationen zur Verfügung, der gleichzeitig viele Objekte für eine Operation kennzeichnen kann. Daher ist die Gewinnung von Datenmengen, wie z.B. Daten in einer Tabelle, nicht effektiv und mit Erhöhung des Netzwerkverkehrs verbunden. Das Fehlen der Eigenschaft, auf große Datenmengen zuzugreifen, würde SNMP nur für kleine Netzwerke eignen, wo der Manager nicht für Netzwerkmanagement auf große Datenmengen angewiesen ist. Um dem SNMP die Fähigkeit zu geben, effektiv auf große Datenmengen zuzugreifen, wurde ein Parallelalgorithmus zur Gewinnung von großen Datenmengen entwickelt [RoMc90b].

Im Gegensatz zu SNMP besitzt CMIP die Fähigkeit, auf große Datenmengen durch die M-GET-Operation zuzugreifen, die ihrerseits die Mechanismen Bereichsbestimmung

---

[40] siehe Seite 184 in [Fish91]

*(scoping)*, Filterung *(filtering)*, verbundene Antworten *(linked replies)* und Synchronisation *(synchronization)* zur Objektauswahl und Gewinnung von Datenmengen benutzt. Daher eignet sich CMIP für große Netzwerke und würde für kleine Netzwerke dem Manager viele zusätzliche Informationen liefern, woran er nicht interessiert ist und so zur unnötigen Erhöhung des Netzwerkverkehrs beitragen.
Die obigen Mechanismen wurden im Rahmen von CMIS beschrieben. Weiter unten soll nur auf die Informationsgewinnung durch SNMP eingegangen werden, indem Beispiele des Zugriffs auf Datenelemente und -mengen den Weg zum Parallelalgorithmus verdeutlichen.

„Die Management-Informationen in der MIB können grob in die innerhalb einer konzeptionellen Tabelle existierenden spaltenförmigen Objekte *(columnar objects)* und die nicht-spaltenförmigen Objekte *(non-columnar objects)* unterteilt werden" [Rose91b]. Die Elemente eines spaltenförmigen Objekts sind die dem Objekt zugeordneten Objektinstanzen, die durch den ASN.1-Objektbezeichner *(ASN.1 Object Identifier)* gekennzeichnet werden. Um eine Objektinstanz eines nicht-spaltenförmigen Objekts zu bezeichnen, wird eine Null dem Objektnamen zugefügt. Als Beispiel kann die einzige Objektinstanz des SysDesc-Objekts, die in einem verwalteten Knoten existiert und wie folgt bezeichnet wird, genannt werden:

```
SysDesc.0      oder      1.3.6.1.2.1.1.1.0 .
```

Für jedes spaltenförmige Objekt in einer Tabelle existiert eine Beschreibung in Textform in der MIB der jeweiligen Tabelle, die den Aufbau des Objektbezeichners beschreibt. Die Objektinstanzen der Spalten der Tabelle *interfaces* werden z.B. durch die Werte der IfIndex-Spalte bezeichnet. Daher ist die mit der ersten Schnittstelle verbundene Objektinstanz des ifDesc-Objekts wie folgt gekennzeichnet:

```
ifDesc.1      oder      1.3.6.1.2.1.2.2.1.2.1 .
```

Durch die Benutzung von Objektbezeichnern zur Benennung von Objektinstanzen werden alle Objektinstanzen einer lexikographischen Ordnung *(lexicographic ordering)* unterworfen.
Während CMIP die komplexen Mechanismen Bereichsbestimmung und Filterung für die Objektwahl benutzt, werden im SNMP von einem verwalteten Knoten unter Berücksichtigung der lexikographischen Ordnung nur die Unterstützung von den drei folgenden einfachen Bedingungen verlangt. Wenn beispielsweise X und Y die Objektnamen sind, dann können sie miteinander wie folgt in Beziehung stehen:

$$X < Y \quad , \quad X > Y \quad \text{oder} \quad X = Y \, .$$

Die Operation *get-next* kann zur Gewinnung von spaltenförmigen und nicht-spaltenförmigen Objekten und ihren Werten benutzt werden. Sie kann auf ein oder mehrere Operanden (Objektbezeichner) angewendet werden und liefert für jeden den Namen und den Wert der direkt darauf folgenden Objektinstanz. Um z.B. die ganze Tabelle *ipRoutingTable* zu durchlaufen, kann mit dem folgenden Aufruf begonnen werden:

```
get-next (ipRouteDest) .
```

Und die Antwort kann dann wie folgt aussehen:

```
ipRouteDest.0.0.0.0 .
```

Wird das Resultat des vorigen Aufrufs als Operand für den nächsten Aufruf verwendet, so kann jede Instanz der Spalte und jede Spalte der Tabelle nacheinander durchlaufen werden. Die Tabelle wird in der Spalte-Reihe-Ordnung durchlaufen, d.h. zuerst jede Instanz der ersten Spalte, dann jede Instanz der zweiten Spalte, usw. bis die letzte Instanz der letzten Spalte aufgefunden ist. Die Tabelle ist dann durchlaufen, wenn auf einen Aufruf eine Antwort zurückkommt, die außerhalb der Tabelle ist, d.h. wenn beispielsweise auf den Aufruf

```
get-next (ipRouteMask.192.33.4.0)
```

die folgende außerhalb der Tabelle liegende Antwort empfangen wird:

```
ipNetToMediaIfIndex.192.33.4.1 .
```

Die Anzahl der benötigten Operationen zur Gewinnung der ganzen Tabelle ist gleich dem Produkt der Anzahl der Zeilen und der Spalten der Tabelle.
Die Anzahl der aufgerufenen Operationen zum Durchqueren einer Tabelle kann reduziert werden, indem der Operation *get-next* anstelle eines Operanden viele Operanden übergeben wird. Dabei ist zu berücksichtigen, daß die Operation *get-next* sequentiell arbeitet und die gleichzeitige Übergabe von Operanden nicht das gleichzeitige Auffinden von Objektinstanzen bedeutet. Das Durchqueren einer Tabelle kann jetzt effektiver stattfinden, da als Operand der Operation *get-next* eine Reihe einer Tabelle übergeben werden kann. Um z.B. die ganze Tabelle *ipRoutingTable* zu durchqueren, wird mit dem folgenden Aufruf begonnen:

```
get-next (ipRouteDest,
          ipRouteIfIndex,
              ...
          ipRouteMask).
```

Die Antwort kann wie folgt aussehen:

```
ipRouteDest.0.0.0.0,
ipRouteIfIndex.0.0.0.0,
        ...
ipRouteMask.0.0.0.0 .
```

Werden die Antworten des vorhergehenden Aufrufs als Operanden von neuem Aufruf verwendet, so können die Reihen der Tabelle nacheinander durchlaufen werden. Der nächste Aufruf für die nächste Reihe der Tabelle lautet dann:

```
get-next (ipRouteDest.0.0.0.0,
          ipRouteIfIndex.0.0.0.0,
                  ...
          ipRouteMask.0.0.0.0).
```

Eine Spalte der Tabelle ist durchlaufen, wenn für diese Spalte eine Antwort zurückkommt, die nicht dem Typ dieser Spalte angehört. Die Tabelle wird dann durchlaufen sein, wenn alle Spalten der Tabelle durchlaufen sind.
Zum Durchqueren eines Abschnitts einer Tabelle sind die Operanden genauer zu spezifizieren. Um z.B. alle Ziele mit der IP-Adresse 128 in der obigen Tabelle zu suchen, kann mit dem folgenden Aufruf gestartet werden:

```
get-next (ipRouteDest.128,
          ipRouteIfIndex.128,
                  ...
          ipRouteMask.128).
```

Eine viel effektivere Methode zur Gewinnung von Informationen einer Tabelle stellt der Parallelalgorithmus zur Verfügung, der in [RoMc90b] beschrieben ist. Der Algorithmus arbeitet wie folgt in zwei Stufen:

- In der ersten Stufe werden alle Objektinstanzen eines spaltenförmigen Objekts, wie z.B. *ipRouteNextHop*, durchlaufen, indem durch Unterteilung der Objektinstanzen in Gruppen, Bereiche festgelegt werden und als Operanden der Operation *get-next* wie folgt übergeben werden [Rose91b]:

  ```
  get-next (ipRouteNextHop,
            ipRouteNextHop.63,
            ipRouteNextHop.127,
            ipRouteNextHop.159,
            ipRouteNextHop.192).
  ```

  Der Algorithmus enthält Informationen über die unbearbeiteten Aufrufe und ihre aktiven Bereiche. Auf eine empfangene Antwort wird reagiert, indem der jeweilige festgelegte Bereich überprüft wird. Liegt die Antwort außerhalb des festgelegten Bereichs, so wird sie entfernt. Anderenfalls wird der Name und der Wert der Objektinstanz gespeichert. Abhängig von der Leistungskapazität der verwendeten Knoten und des Netzwerks können weitere Bereichsunterteilungen vorgenommen werden.
  Ein spaltenförmiges Objekt ist durchlaufen, wenn alle Bereiche durchlaufen und entfernt worden sind.

- In der zweiten Stufe werden die Objektinstanzen eines anderen spaltenförmigen Objektes in der gleichen Tabelle gewonnen, indem jetzt die Operation *get* benutzt wird und als Operanden die Ergebnisse der ersten Stufe verwendet werden.

„Dieser Algorithmus ist in einer Prototyp-Umgebung implementiert worden und zeigt gegenüber der seriellen Informationsgewinnung eine Beschleunigung von annähernd einer Größenordnung an“[41].

Während SNMP erst durch zusätzliche Hilfsmittel, wie den oben beschriebenen Parallelalgorithmus, auf Datenmengen effektiv zugreifen kann, stehen dem CMIP nach Definition Hilfsmittel, wie die Mechanismen Bereichsbestimmung und verbundene Antworten, zur Verfügung. Die neuen verwalteten Objekte in der MIB können z.B. mittels Objektauswahlmechanismen im CMIP effektiver und schneller im Vergleich zum SNMP, das die Durchquerung des MIB-Baumes hierfür benötigt, gefunden werden. CMIP erlaubt einer Anwendung z.B. große Datenmengen von verschiedenen verwalteten Systemen über eine einzelne Anfrage zu gewinnen, wobei die Daten in mehreren verbundenen Antworten zurückgesendet werden. SNMP unterstützt nicht den Mechanismus der verbundenen Antworten und ist auch deswegen bei der Informationsgewinnung beschränkt. Die für CMIP definierten Mechanismen[42] können z.B. innerhalb der M-Get-Operation zur besseren Objektauswahl und effektiveren Informationsgewinnung benutzt werden und sind als Vorteil für CMIP anzusehen. Der bezüglich der verwalteten Knoten im Vergleich zur einfachen SNMP-Strategie nachteilig wirkende Implementationsaufwand dieser Mechanismen soll ebenfalls berücksichtigt werden.

### 4.5.2 Zyklisches Abfragen vs. Berichten

Eine wesentliche Aufgabe des Netzwerkadministrators besteht darin, den Kommunikationsdienst jederzeit mit ausreichender Qualität zur Verfügung zu stellen. Zur Erfüllung dieser Aufgabe wird ein Netzwerkmanagement-Mechanismus benötigt, der dem Netzwerkadministrator Informationen über den aktuellen Betriebszustand des Netzwerks und die auftretenden Ereignisse im Netzwerk liefert. Ein Netzwerkadministrator kann dann rechtzeitig die entsprechenden Entscheidungen treffen, um den einwandfreien Betrieb des Netzwerks zu sichern.

Die Diskussion über Vorteile und Nachteile der beiden Mechanismen, Zyklisches Abfragen *(polling)* und Berichten *(reporting)*, hat seinen Ursprung in der entsprechenden Diskussion über Betriebssysteme, wo die Verfahren Unterbrechung *(interrupt)* und Zyklisches Abfragen miteinander verglichen werden.

Unter ***Zyklischem Abfragen*** wird das regelmäßige Abfragen von verwalteten Knoten nach ihren Zuständen durch die Management-Station oder NOC[43] verstanden. Durch Zyklisches Abfragen wird der Management-Station ein relativ reales Bild vom Netzwerk geliefert. Ein Maß für die Aktualität der der Management-Station gelieferten Zustandsinformationen ist die Abfragerate. Eine niedrige Abfragerate hat den Nachteil,

---

41 siehe Seite 19 in [Rose91b]

42 siehe auch „2.5.5.1 Allgemeiner Management-Informationsdienst“ auf Seite 39

43 Der Satz von durch Administratoren eines großen Netzwerks betriebenen Management-Stationen wird Netzwerk-Operationszentrum *(Network Operations Center, NOC) genannt* [*Rose91a*].

daß auf aufgetretene Fehler spät reagiert werden kann, während eine hohe Abfragerate zusätzliche Bandbreite in Anspruch nimmt, was die Leistungsfähigkeit des Netzwerks reduziert [Mard91]. Aus diesem Grund ist es die Aufgabe des Administrators, die beste Lösung für sein Netzwerk bezüglich der Abfragerate zu finden. Eine weitere Möglichkeit zur Verminderung der Bandbreitevergeudung stellt das Abfragen von nur kritischen Objekten dar. Ein kritisches Objekt ist ein verwaltetes Objekt mit der Eigenschaft, daß seine Fehlfunktion katastrophale Folgen für den Netzwerkbetrieb hat.

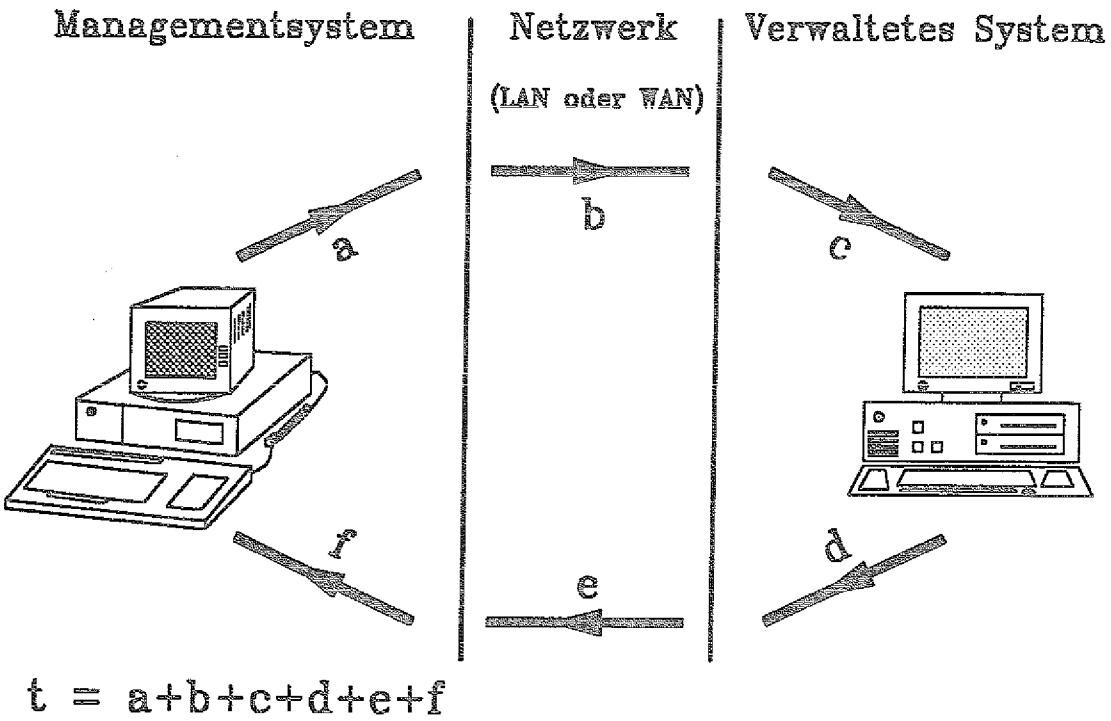



**Abb. 34. Zyklisches Abfragen:** Beim Zyklischen Abfragen holt der Manager im Managementsystem periodisch Informationen über die zu verwaltenden Geräte. Die Informationen werden ihm durch die Agenten in den verwalteten Systemen bereitgestellt. Dabei ist die Zeit $t$ für einen Abfragezyklus gleich der Summe von:

- $a$: Initiierungszeit einer Anfrage durch den Manager,
- $b$: Netzwerkverzögerungszeit vom Manager zum Agenten,
- $c$: Empfangs- und Interpretationszeit des Agenten,
- $d$: Beantwortungszeit des Agenten,
- $e$: Netzwerkverzögerungszeit vom Agenten zum Manager und
- $f$: Empfangs- und Interpretationszeit des Managers.

Durch das Zyklische Abfragen kann nur eine begrenzte Anzahl von Objekten verwaltet werden. Ist die minimal notwendige Zeit für die Durchführung einer einzelnen Abfrage $t$ und die Zeitspanne zwischen zwei Abfragen $T$, so läßt sich nach einem einfachen Modell die Anzahl $N_p$ der Objekte, die verwaltet werden können, wie folgt berechnen:

$$N_p \leq \frac{T}{t} \qquad (1).$$

Unter dem *Berichten* versteht man einen Mechanismus, mit dessen Hilfe ein verwaltetes Objekt die Management-Station oder den Manager, der Management-Station entsprechende Instanz im OSI-Netzwerkmanagement, vom Auftreten eines Ereignisses benachrichtigt. Daher ist der Vorteil dieses Verfahrens die sofortige Benachrichtigung des Managers, was beim Zyklischen Abfragen eine Funktion der Abfragerate ist. Es ist aber zu berücksichtigen, daß die Voraussetzungen für die Benachrichtigung des Managers die Funktionsfähigkeit des verwalteten Objekts, eine Verbindung zwischen dem verwalteten Objekt und dem Manager und die Aufnahmebereitschaft des Managers sind. Die Nachteile des Berichtens können wie folgt zusammengefaßt werden:

- hohe Forderungen an die verwalteten Knoten, da sie jetzt Ressourcen für die Erzeugung von Ereignismeldungen besitzen müssen,
- Verkleinerung des Wirkungsgrads von verwalteten Knoten, da sie jetzt zeitweise statt ihrer eigentlichen Aufgabe mit Fehlerbehandlung beschäftigt sind, und
- Inanspruchnahme einer großen Bandbreite, wenn keine Schwelle *(threshold)* für die dem Manager zu sendenden Ereignismeldungen gesetzt wird, was zusätzliche Ressourcen und längere Bearbeitungszeit für das verwaltete Objekt bedeutet.

Im Unterschied zum Zyklischen Abfragen, mit dem die Verwaltung einer beschränkten Anzahl von Objekten möglich ist, wird das Berichten durch die Geschwindigkeit, mit der ein Manager die Meldungen verarbeiten kann, begrenzt. Erreichen die Meldungen den Manager mit der Rate $\omega$ und wird für die Verarbeitung einer einzelnen Meldung die Zeit $\tau$ benötigt, so kann die Anzahl $N_r$ der Objekte, die durch das Berichten verwaltet werden können, wie unten dargestellt berechnet werden:

$$N_r \leq (\frac{1}{\omega})(\frac{1}{\tau}) \qquad (2).$$

Zur Vereinfachung der Rechnung soll angenommen werden, daß die Zeit zwischen zwei Abfragen $T$ beim Zyklischen Abfragen und die Zeit zwischen zwei, vom Manager empfangenen Meldungen $\frac{1}{\omega}$ beim Berichten gleich sind, d.h.

$$T = \frac{1}{\omega} \qquad (3),$$

obwohl im OSI-Netzwerkmanagement wegen der eingebauten Schwellen für die Meldungen nicht jedes Ereignis zur Entlastung des Netzwerkverkehrs gemeldet wird und dadurch $T < \frac{1}{\omega}$ ist.
Die Zeit zur Durchführung einer Abfrage $t$ setzt sich nach diesem Modell aus der Verarbeitungszeit durch den Manager $a+f$, der Verzögerungszeit durch das Netzwerk $b+e$ und die Verarbeitungszeit durch das verwaltete System $c+d$ zusammen, d.h.

$$t = a + b + c + d + e + f$$

(vgl. Abb. 34). Unter der Annahme, daß der Manager im Berichten wie im Zyklischen Abfragen für eine Meldung die Verarbeitungszeit $\tau = f$ benötigt, ist die Aussage

$$\tau < t \qquad (4)$$

richtig. Aus (1), (2), (3) und (4) folgt:

$$N_p \leq N_r \ .$$

Das obige nach einem vereinfachten, in [BeCW90] beschriebenen Modell berechnete Ergebnis zeigt, daß mittels Zyklischen Abfragens verglichen mit dem Berichten weniger Objekte verwaltet werden können. Während $N_p$ wegen der Abfragezeit für einen Zyklus $t$ der Verzögerungszeit des Netzwerks umgekehrt proportional ist, ist $N_r$ von der Verzögerungszeit innerhalb des Netzwerks unabhängig.

SNMP benutzt ein trap-orientiertes Zyklisches Abfragen *(trap-oriented polling)* zur Erkennung der aufgetretenen Ereignisse. Ein Trap ist die im SNMP-Netzwerkmanagement verwendete Bezeichnung für ein Ereignis *(event)*, das durch einen Agenten im verwalteten Objekt dem Manager geschickt wird. Aufgrund der oben genannten Nachteile vom Berichten werden Traps im SNMP-Netzwerkmanagement sehr selten benutzt. Daher wurden nur sechs Trap-Typen, die die folgenden Ereignisse anzeigen, innerhalb des SNMP-Netzwerkmanagements definiert[44]:

1. Initialisierung eines Agenten,
2. Wiederanlauf eines Agenten,
3. Ausfall einer Verbindung,
4. Wiederanlauf einer Verbindung,
5. schlechte Authentifizierung und
6. Verlust eines EGP-Nachbars, wobei unter EGP-Nachbar jeder Router, der mit einem einzelnen Router über EGP kommuniziert, verstanden werden soll.

Um die Erweiterbarkeit, die eine wichtige Eigenschaft einer guten Netzwerkmanagement-Architektur ist, im SNMP-Netzwerkmanagement nicht zu beschränken, wurde den Herstellern die Definition eigener für sie als signifikant bezeichnete Traps erlaubt.
Während durch das SNMP-Netzwerkmanagement eine begrenzte Anzahl von Traps definiert worden ist und eine Erweiterung der Trap-Anzahl nach Notwendigkeit durch den Hersteller nicht ausgeschlossen ist, definiert das OSI-Netzwerkmanagement bei der Normierung eine große Anzahl von wahlfreien und obligatorischen Ereignissen [FeRS90].

Das trap-orientierte Zyklische Abfragen stellt einen Kompromiß zwischen den beiden Mechanismen Zyklisches Abfragen und Berichten dar und läuft wie folgt ab: Nach dem Auftreten eines außerordentlichen Ereignisses wird durch das verwaltete Objekt ein einzelner, einfacher Trap zur Management-Station geschickt, wie es beim Berichten der Fall ist. Danach ist die Management-Station für die Aktivierung weiterer Maßnahmen zuständig, die Art und Ausmaß des Problems bestimmen. Da Traps im SNMP durch einen verbindungslosen Transportmechanismus geschickt werden, sollen sie als eine frühe Warnung betrachtet werden. Zur Sicherstellung des Netzwerkbetriebs werden die verwalteten Knoten zusätzlich mit niedriger Abfragerate zyklisch abgefragt [Rose91b]. Durch den obigen Kompromiß werden die folgenden Ziele erreicht:

- Forderungen an verwalteten Objekten bleiben minimal, da nur einzelne einfache Traps geschickt werden müssen, d.h. das fundamentale Axiom ist erfüllt;

---

[44] siehe auch „3.6.3.6 Traps" auf Seite 101

- Vergeudung der Netzwerkbandbreite wird verringert, da mit niedriger Abfragerate abgefragt wird;
- Probleme können im akzeptablen Zeitrahmen behandelt werden, da die Abhängigkeit von der Abfragerate für das Entdecken eines Problems reduziert worden ist.

Während SNMP das trap-orientierte Zyklische Abfragen verwendet, benutzt CMIP den Berichten-Mechanismus, um dem Manager detaillierte Informationen über ein auftretendes Ereignis zu vermitteln. Die Ereignismeldungen im CMIP können bestätigt oder unbestätigt sein. Im SNMP dagegen sind die Ereignismeldungen nur unbestätigt. Zur Normierung dieser dem Manager zugesandten Ereignismeldungen steht die Alarmmeldungs-Funktion [ISO10164-4][45] zur Verfügung, die einen allgemeinen Satz von Meldungen mit Parametern und ihren Werten definiert. Außerdem muß der Strom dieser Meldungen mit unterschiedlichen Dringlichkeitsgraden gesteuert werden, falls die Anzahl der dem Manager zugesandten Meldungen begrenzt bleiben soll. Die Ereignisbericht-Management-Funktion [ISO10164-5] stellt die dazu notwendigen Hilfsmittel zur Verfügung. Die Meldungen über Ereignisse müssen aus verschiedenen Gründen gespeichert werden können. Die Log-Steuerungs-Funktion [ISO10164-6] definiert Komponenten und ihre Steuerung für die Informationsspeicherung in einem offenen System.

CMIP basiert im Gegensatz zum SNMP, das einen verbindungslosen Transportmechanismus benutzt, auf einem verbindungsorientierten Transportmechanismus, d.h. es benötigt für die Übertragung von Meldungen eine Verbindung zwischen dem Manager und dem verwalteten Objekt. Soll das verwaltete Objekt auf Bestätigungen des Managers warten, so muß eine Verbindung aufrechtgehalten werden oder jedesmal eine neue Verbindung aufgebaut werden. Hierdurch werden die verwalteten Knoten mit zusätzlichen Aufgaben belastet, was ihren Wirkungsgrad nachteilig beeinflußt.

### 4.5.3 Funktionalität

SNMP, das als eine kurzfristige Lösung zum Netzwerkmanagementproblem entworfen wurde [Cerf88], spezifiziert ein einfaches Netzwerkmanagement-Protokoll, das die wichtigsten Forderungen an Netzwerkmanagement erfüllen kann und ist so erweiterbar, daß zukünftige und unberücksichtigte Aspekte vom Netzwerkbetrieb und -management realisiert werden können [CFSD90a]. Die Konzentration auf die wesentlichen Aufgaben des Netzwerkmanagements ist eine Forderung des Einfachheitsprinzips, das die Grundlage der Entwicklung im SNMP darstellt. Deshalb verfügt SNMP nur über die Mechanismen, die wirklich für die Überwachung und Steuerung von realen Netzwerken erforderlich sind.

Im Gegensatz dazu wurde CMIP als das Netzwerkmanagement-Protokoll, das der Überwachung und Steuerung der komplexen zukünftigen Netzwerke dienen soll, mit Betonung auf die Globalität und Vollkommenheit entworfen. Um allen möglichen Anforderungen gerecht zu werden, mußte das Protokoll mit vielen spezifischen Eigenschaften und Fähigkeiten ausgestattet werden. Einige wesentliche Beispiele sind:

---

[45] siehe „2.5.6.4 Alarmmeldungs-Funktion“ auf Seite 56

- objektorientiertes Modell,
- effiziente Objektauswahl,
- effektive Informationsgewinnung,
- Sicherheitsaspekte und
- vielfältige Funktionen innerhalb des System-Managements.

CMIP stellt die Möglichkeit zur Verfügung, komplizierte bedingte Kommandos zu implementieren, die mit Objekttyp, -wert oder relativer Position im Netzwerk operieren. Beispiele sind Objektauswahl[46] mittels Mechanismen Bereichsbestimmung *(scoping)* und Filterung *(filtering)*. Im Unterschied dazu sind im SNMP die obigen Fähigkeiten nicht vorhanden, und die Objekte müssen für eine Operation genau spezifiziert werden. Eine Ausnahme stellt die Operation *get-next* dar, die das lexikographisch nächste Objekt in der MIB zurückgibt [Pres90].
Eine für einen verwalteten Knoten spezifizierte Managementanwendung, wie z.B. eine Router-Managementanwendung, in einem auf CMIP-basierenden Managementsystem ist in der Lage, alle eine Bedingung erfüllenden Router auszuwählen. Im Vergleich hierzu muß eine SNMP benutzende Anwendung für die gleiche Aufgabe erstens über alle Router im Netzwerk Bescheid wissen, zweitens die spezielle Informationen für alle Router holen und drittens prüfen, für welche Router die Bedingung erfüllt ist.

SNMP und CMIP stellen die Fähigkeit zur Verfügung, mehrere Operationen vom gleichen Typ innerhalb einer Anfrage dem verwalteten Knoten zu übergeben. Diese Fähigkeit wird Synchronisation[47] *(synchronization)* genannt, die unteilbar *(atomic)* und teilbar *(best effort)* durchgeführt werden kann. Während CMIP die beiden Synchronisationsarten unterstützt, ist im SNMP nur die unteilbare Synchronisation möglich. Die Unteilbarkeit ist so zu verstehen, daß wenn eine einzelne Operation aus der Menge der Operationen in einer Anfrage nicht ausgeführt werden kann, keine der Operationen ausgeführt und eine Fehlermeldung zurückgeschickt wird. Die Operation *set* mit mehreren Variablen im SNMP stellt ein Beispiel für die unteilbare Synchronisation dar. Wird eine der durch *set* initiierten Änderungen aus irgendeinem Grund nicht durchgeführt, so wird keine der Änderungen durchgeführt. Dieses Verhalten sichert z.B., daß ein Eintrag in allen Routing-Tabellen oder in keiner Routing-Tabelle geändert wird.

CMIP besitzt im Gegensatz zum SNMP imperative Kommandos, die die Erzeugung von Objekten oder die Ausführung von Aktionen erlauben. CMIP kann z.B. durch ein Kommando das System neu booten. SNMP realisiert aber diese Fähigkeit durch Überlisten der Architektur, indem ein Parameter in der MIB gesetzt wird, der das System zum Neubooten zwingt.

Im OSI-Netzwerkmanagement werden mehrere Funktionen[48] innerhalb des System-Managements definiert, die die CMIS-Dienste in Anspruch nehmen und zur Erledigung von Managementaufgaben benutzt werden. SNMP definiert keine äquivalenten Funk-

---

46 siehe „2.5.5.1 Allgemeiner Management-Informationsdienst“ auf Seite 39

47 siehe „2.5.5.1 Allgemeiner Management-Informationsdienst“ auf Seite 39

48 siehe „2.5.6 Beispiele für System-Management-Funktionen“ auf Seite 48

tionen. „Es wird erwartet, daß zukünftige Funktionen innerhalb des System-Managements CMIP für die Unterstützung der Manager-zu-Manager-Kommunikation besser unterstützen werden. SNMP unterstützt die Wechselwirkungen, die spezifisch zur Manager-zu-Manager-Kommunikation sind, nicht ausreichend" [BeCW90].

Die Brauchbarkeit und Notwendigkeit von spezifischen Fähigkeiten wird von vielen der SNMP-Befürworter abgelehnt. Beispielsweise wird die Fähigkeit vom CMIP, Gewinnung von großen Datenmengen oder Tabellen, mit der Argumentation als unbrauchbar bezeichnet, daß ein Netzwerkadministrator sich in erster Linie für einen Teil der Tabelle interessiert und nicht für die ganze Tabelle [Bors90].
Andere akzeptieren diese spezifischen Eigenschaften und Fähigkeiten innerhalb CMIP und behaupten, daß „die meisten dieser Eigenschaften sehr elegant mittels SNMP realisiert werden können, wenn man sich nur in SNMP gut auskennt"[49].

### 4.5.4 Speichergröße und Leistung

Die Untersuchung von SNMP und CMIP auf der Basis ihrer Standards und Modelle läßt bezüglich Speichergröße und Rechenleistung folgern, daß SNMP weniger Speicherplatz und Rechenleistung als CMIP benötigt und deshalb billiger zu realisieren ist.
SNMP stellt nach dem fundamentalen Axiom minimale Forderungen an die verwalteten Objekte und benötigt deswegen wenige zusätzliche Ressourcen, die nur zur Erfüllung der Managementaufgaben erforderlich wären. SNMP basiert auf dem Internet-Protokoll *(Internet Protocol, IP)* und benutzt ein einfaches Transportprotokoll, nämlich das Benutzer-Datagramm-Protokoll *(User Datagram Protocol, UDP)*, das einfach der Anzeige einer SNMP-Dateneinheit in der Nachricht dient. Die Tatsache, daß SNMP auf einem minimalen Protokollturm operiert und die Protokollverarbeitung selbst einfach ist, ermöglicht eine Reduzierung der in einem verwalteten Knoten zu implementierenden Software, die auch Agent *(agent)* genannt wird. Der Einsatz minimaler Soft- und Hardware in verwalteten Knoten ist eine der größten Vorteile von SNMP [Herm91, Robe90, Benn89, Rose91c].

Um die Größe der von SNMP benötigten Speicherkapazität zu veranschaulichen, wurde SNMP 1990 in der Interop-Ausstellung[50] zur Steuerung und Überwachung von alltäglichen Geräten und Spielzeug, d.h. einem Toaster, einem CD-Player und einem batteriebetriebenen bellenden Hund verwendet. Der Toaster wurde über das Netz durch ein SNMP-Programm, das auf einem PC lief, gesteuert. Der CD-Player hatte eine X-Window-System-Schnittstelle, mit deren Hilfe man die verfügbaren Discs aus einer Bibliothek, die zu spielenden Lieder und die Lautstärke über das Netz bestimmen konnte. Der CD-Player wurde durch einen nicht gerätespezifischen sogenannten Schwarzen Kasten mit dem Netz verbunden, der mit nur 64KBytes realisiert worden war. Der batteriebetriebene bellende Hund konnte von zwei Manager-Konsolen gesteuert werden, wobei eine gleichzeitige Steuerung zu einer Fehlermeldung führte [Fish91].

---

49 siehe Seite 186 in [Fish91]

50 Eine jährlich stattfindende Ausstellung und fachliche Konferenz für Aufbau und Management heterogener Netze

Durch die Verlagerung der Verantwortung von verwalteten Knoten zu den Management-Stationen sind die verwalteten Knoten in der Lage, sich mit ihren eigentlichen Aufgaben zu beschäftigen und müssen nicht einen großen Teil ihrer Leistung zum Zweck des Managements zu vergeuden. Dies bedeutet, daß ein verwalteter Knoten mehr MOPS[51] liefern kann, wenn er durch SNMP statt CMIP gesteuert wird.

Eine Folge der obigen Überlegungen ist, daß SNMP keine teuere Technologie im Vergleich zum CMIP zur Verfügung stellt. Diese Aussage verliert ihre Gültigkeit, wenn die Netzwerkgröße und -komplexität zunimmt, da die Kosten für SNMP schneller als für CMIP ansteigen (Abb. 35). Aus diesem Grund eignet sich SNMP für LANs *(Local Area Networks)* und einfache, miteinander verbundene Netze, was wiederum seine starke Verbreitung in solchen Umgebungen begründet [CrWa91].

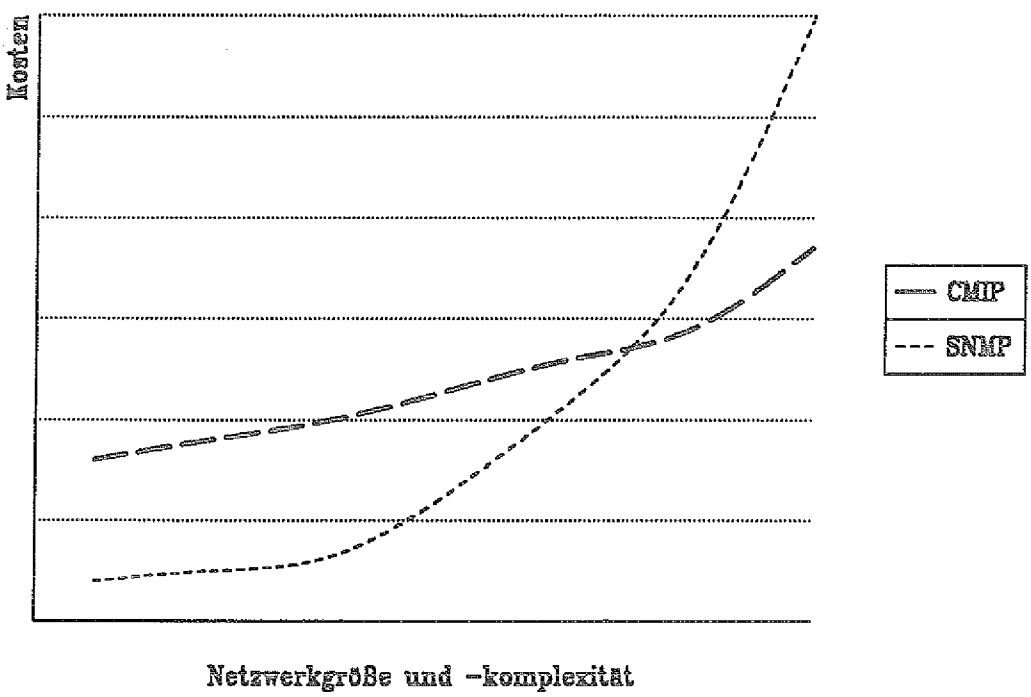



**Abb. 35. Netzwerkgröße und -komplexität** [CrWa91]: SNMP ist in einfachen verbundenen Netzwerken und LANs billiger als CMIP zu implementieren, während sich CMIP für größere und komplexere Netzwerke aufgrund des langsameren Kostenanstiegs im Vergleich zu SNMP eher eignet.

Im Gegensatz zu SNMP ist CMIP ein kompliziertes Netzwerkmanagement-Protokoll, das auf den sieben Schichten des OSI-Protokollturms basiert, und stellt deshalb große

---

51 Management-Operationen pro Sekunde *(management operations per second)* ist ein normiertes Maß für CPU-Benutzung im Netzwerkmanagement. Da MOPS vom systemspezifischen Parameter MIPS *(millions of instructions per second)* abhängig ist, ist das Verhältnis MOPS/MIPS ein angemessenes maschinenunabhängiges Benchmark für die Netzwerkmanagementprotokolle [BeCW90].

Forderungen an die verwalteten Knoten bezüglich Speicherkapazität und Rechenleistung. Das Problem besteht also in der Integration der ganzen OSI-Komplexität in jedem Netzwerkelement, das verwaltet werden soll [Herm91, Rose91c]. Ein anderes Problem ist, daß zur Zeit nicht viele Netzwerke existieren, die alle Protokolle des OSI-Protokollturms implementieren [Jand91a].

Die Verlagerung der Verantwortung vom Manager zu den verwalteten Objekten spezifiziert die Vorgehensweise von CMIP bezüglich Aufgabenverteilung und kann beispielhaft durch den Mechanismus Berichten im Vergleich zum Zyklischen Abfragen verdeutlicht werden. Die verwalteten Objekte müssen deshalb zusätzliche Ressourcen besitzen, um die spezifischen Fähigkeiten vom CMIP realisieren zu können. Obwohl einige das Problem des Ressourcenaufwands in jedem verwalteten Objekt mit ständigem Abfall der Hardwarepreise als gelöst bezeichnen, findet diese Argumentation keine Unterstützung aus der Vergangenheit. Die Vergangenheit hat gezeigt, „wie die Abnahme von Verarbeitungskosten/Leistung es attraktiver machte, existierende Fähigkeiten den kleineren Geräten zuzufügen, statt die Geräte mit neuen Eigenschaften auszurüsten“[52].

Die hohe Forderung an den verwalteten Knoten infolge der Implementierung des gesamten OSI-Protokollturms und der Seltenheit von OSI-Netzwerken haben die Hersteller dazu geführt, CMIP über andere Protokolle laufen zu lassen [Bors90, Bona90b]. Die wesentlichen Beispiele hierfür sind:

- CMIP über SNA *(Systems Network Architecture)* von IBM,
- CMIP über TCP/IP *(CMIP over TCP/IP, CMOT)*,
- CMIP über einen verbindungslosen OSI-Transportmechanismus und
- CMIP direkt über LLC *(Logical Link Control)* von IEEE 802 *(CMIP over LLC, CMOL)*.

Im folgenden werden CMOL, als nahe Alternative für auf sieben Schichten basierendem CMIP, und CMOT, als langfristige Lösung für Management von auf TCP/IP-basierenden Netzwerken, beschrieben.

#### 4.5.4.1 CMIP über LLC (CMOL)

Das Netzwerkmanagement-Protokoll CMOL *(CMIP over LLC)* ist von den Herstellern IBM und 3Com im Rahmen der Spezifikationen für das Management heterogener LAN-Umgebungen *(Heterogeneous LAN Management, HLM)* entwickelt worden. HLM und dadurch sein Netzwerkmanagement-Protokoll CMOL basieren auf den zwei ersten Schichten, nämlich der Bitübertragungsschicht und der Sicherungsschicht, wobei die Sicherungsschicht bei lokalen Netzen in zwei Teilschichten LLC und MAC *(Media Access Control)* unterteilt wird (Abb. 36) [VaCa90, Carr90].

Die HLM-Architektur spezifiziert einen LAN-Station-Manager *(LAN station manager)*, der der Bereitstellung von allgemeinen Basis-HLM-Diensten für die

---

[52] siehe Seite 17 in [Rose91c]

HLM-Stationen dient, und einen HLM-Agent, der die Verwaltung jedes ans Netz angeschlossenen Knotens ermöglicht. Schließlich werden drei MIBs spezifiziert[53]:

1. „MAC-Schicht-MIB für Ethernet und Token Ring,
2. LLC-Schicht-MIB und
3. Umgebung- oder Ressource-MIB für das Management von Stationsmerkmalen, wie z.B. DOS-Version".

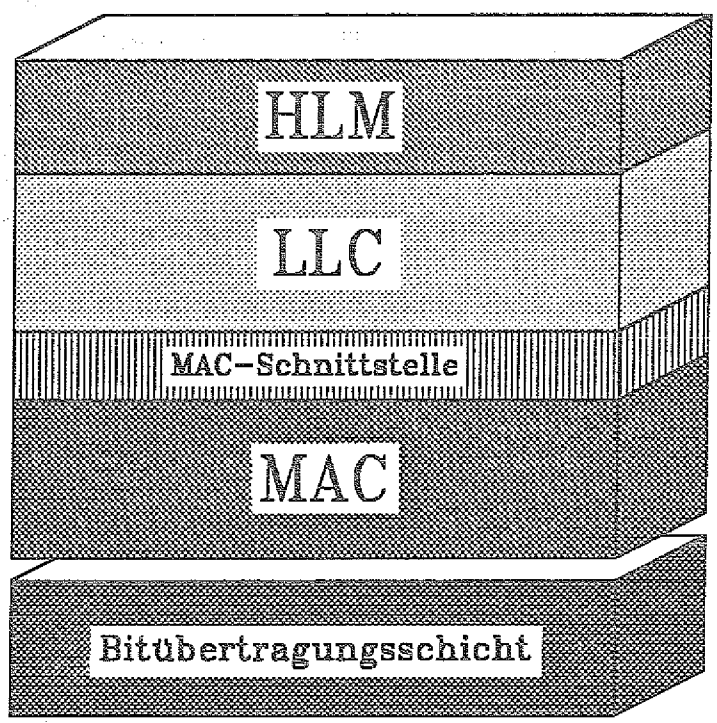



**Abb. 36. CMOL-Architektur:** CMOL, das als Netzwerkmanagement-Protokoll der HLM-Spezifikationen gilt, arbeitet nur mit der Bitübertragungs- und Sicherungsschicht und benötigt deshalb weniger Speicherplatz zur Implementation der darauf basierenden Schichten. Als MAC-Schnittstelle, die Schnittstelle zwischen MAC- und LLC-Teilschicht, können verschiedene Protokoll-Schnittstellen benutzt werden, wie z.B. NDIS *(Network Driver Interface Specification)*, die bei LAN-Manager und VINES verwendet wird, oder ODI *(Open Data link Interface)* von Novell [Doug91].

Das Komitee IEEE 802.1, das für die Definition eines Netzwerkmanagement-Protokolls für LANs zuständig ist, hat sich für die Aufnahme von CMOL in seinen Management-Protokoll-Standard 802.1B entschieden [Doug91]. Das 802.1B-Protokoll soll den Anschluß von Netzwerkgeräten zu den LANs in einer normierten Weise ermöglichen und

---

53 siehe Seite 80 in [Doug91]

dadurch die Implementationskosten in einfachen Geräten oder Geräten mit beschränkter Speicherkapazität minimieren.

CMOL soll ein einheitliches Management-Protokoll für LANs durch Austausch von CMIP-Protokoll-Dateneinheiten über einfache verbindungslose Dienste definieren. Hierdurch sollen die Nachteile beseitigt werden, die dem auf den OSI-Protokollen der sieben Schichten basierenden CMIP zugeordnet werden. Ein durch CMOL verwalteter Knoten braucht in diesem Fall nur den Kode für zwei Schichten statt für sieben Schichten zu implementieren und kann deshalb ein einfaches Gerät wie z.B. ein PC sein. Die direkte Folgerung des von CMOL verwendeten kleinen Protokollturms ist, daß die Prozessorzeit mehr der eigentlichen Aufgabe des Gerätes gewidmet wird, und dadurch CMOL mehr MOPS im Vergleich zum CMIP in einem Gerät zur Verfügung stellen kann [Jack91a].

Die aus der CMOL-Definition resultierende Beschränkung auf die lokalen Netze besteht darin, daß die LANs, die mittels CMOL überwacht und gesteuert werden, nicht durch Routers miteinander verbunden werden können [Simp91]. CMOL operiert nach seiner Definition direkt auf der LLC-Schicht und nicht auf der Vermittlungsschicht. Daher verfügen die Nachrichten über keinen Nachrichtenkopf *(header)* der Vermittlungsschicht, wie z.B. IP-Nachrichtenkopf, der für die Weiterleitung der Nachrichten durch Routers benötigt wird. Die durch CMOL überwachten und gesteuerten LANs werden deshalb durch MAC-Schicht-Bridges miteinander verbunden. Eine andere Möglichkeit besteht in der Benutzung eines Vertretungsagenten *(proxy agent)*, der die empfangenen CMOL-Nachrichten durch einen allgemeineren Transportmechanismus, wie z.B. TCP, SNA oder OSI-Transportprotokolle, weiterleitet.

#### 4.5.4.2 CMIP über TCP/IP (CMOT)

CMOT *(CMIP over TCP/IP)* ist im Rahmen der IAB-Strategie für einen allmählichen Übergang zu den von der ISO definierten Netzwerkmanagementprotokollen entwickelt worden. Die Definition umfaßt einen ausgewählten Protokollsatz zusammen mit den für die Implementierung der Protokolle benötigten Vereinbarungen. Der CMOT-Protokollsatz besteht aus sieben Protokollen (vgl. Abb. 37):

1. ACSE *(ISO)*,
2. ROSE *(ISO)*,
3. CMIP *(ISO)*,
4. LPP,
5. UDP,
6. TCP und
7. IP.

CMOT stellt die durch CMIS für CMIP definierten Anwendungsdienste zur Verfügung, die einer Management-Anwendung mittels CMISE bereitgestellt werden. Die Anwendungsschicht entsprechend dem OSI-Referenzmodell ist in der CMOT-Architektur und der OSI-Netzwerkmanagement-Architektur gleich. Das heißt, daß die Anwendungsschicht unter anderem die Dienst-Elemente CMISE für die Kommunikation mit dem Partnersystem und ACSE und ROSE für die Kommunikation mit den darunterliegenden Schichten umfaßt. Der Übergang von der Anwendungsschicht zur Transport- und Ver-

mittlungsschicht, die in diesem Fall jeweils TCP/UDP und IP implementieren, findet über ein Darstellungsschicht-Protokoll statt, das LPP *(Lightweight Presentation Protocol)* genannt wird. LPP liefert also einen Mechanismus zur Unterstützung der OSI-Anwendungsdienste direkt über die TCP/IP-Umgebung.

Die Vereinbarungen zur Implementierung besagen, daß ein CMOT-konformes System die Protokolle ACSE, ROSE, CMIP, LPP und IP implementieren und die Benutzung von LPP über UDP oder TCP unterstützen muß. Darüber hinaus ist die Unterstützung der Benutzung von LPP über die beiden Protokolle UDP und TCP in gleichem System erlaubt [WBLH90].

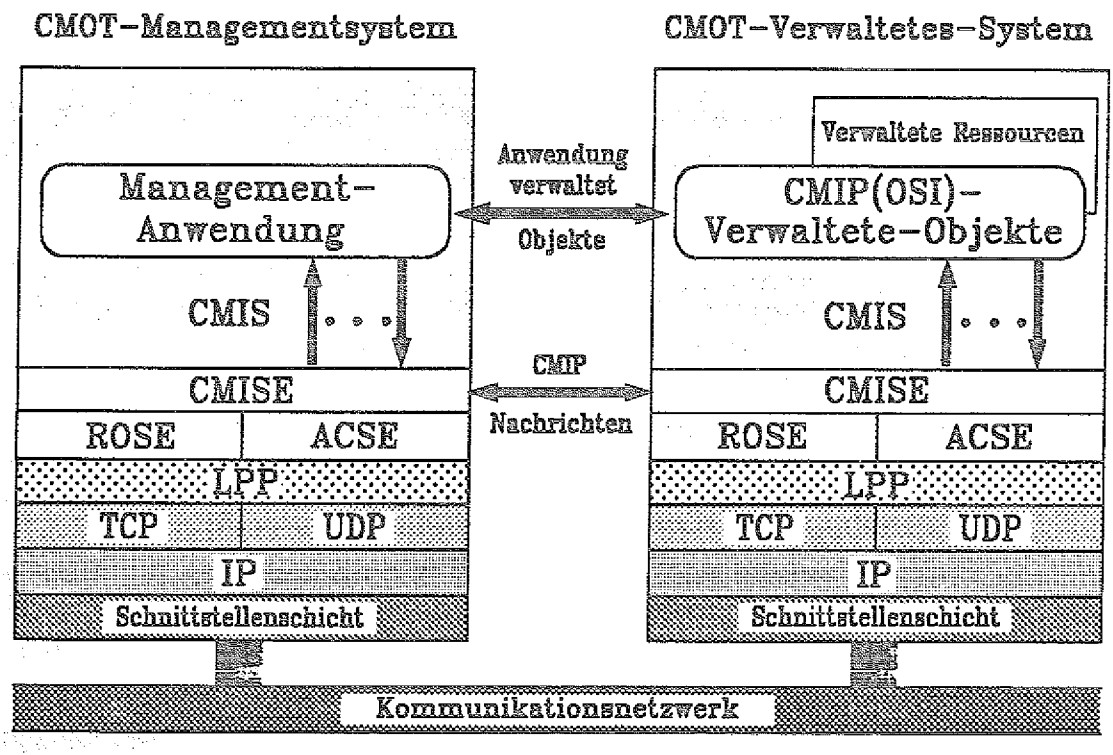



Abb. 37. CMOT-Architektur: Dargestellt sind die Protokolle der CMOT-Architektur und die Wechselwirkungen zwischen dem CMOT-Managementsystem und dem CMOT-Verwaltetes-System. Das Protokoll LPP ermöglicht es, die Anwendungsschichtprotokolle des OSI-Netzwerkmanagements über TCP/IP-Protokolle zu implementieren. So werden die OSI-Anwendungsdienste direkt über die TCP/IP-Umgebung unterstützt.

### 4.5.5 Transportprotokoll

Die Protokolle SNMP und CMIP benutzen zur Übertragung von Informationen zwischen Knoten in einem Netzwerk unterschiedliche Transportdienste bzw. Transportprotokolle. Während SNMP das verbindungslose Transportprotokoll UDP benutzt, verwendet CMIP das verbindungsorientierte Transportprotokoll TP4. So kann SNMP als ein verbindungsloses Protokoll und CMIP als ein verbindungsorientiertes Protokoll aufgefaßt werden. Im folgenden werden die beiden Transportarten und -protokolle kurz

erläutert, damit die aus ihrer Auswahl resultierenden Vor- und Nachteile für das jeweilige Netzwerkmanagement besser verstanden werden können.

Ein einem Netzwerkmanagement-Protokoll unterliegender Transportdienst, wie jeder andere durch eine Schicht gelieferte Dienst in OSI, kann verbindungsorientiert oder verbindungslos sein. Ein verbindungsorientierter Dienst durchläuft drei verschiedene Phasen, die

1. Verbindungsaufbau,
2. Datenübertragung und
3. Verbindungsabbau

genannt werden. Im Unterschied dazu besitzt ein verbindungsloser Dienst nur eine Phase, nämlich die Datenübertragungsphase.
OSI stellt die fünf verbindungsorientierten Transportprotokolle TP0 bis TP4 zur Verfügung. Dabei basieren die ersten vier Protokolle auf einem verbindungsorientierten Vermittlungsdienst, während das TP4 sowohl verbindungsorientierten als auch verbindungslosen Vermittlungsdienst unterstützt.
Internet definiert die Transportprotokolle TCP, ein verbindungsorientiertes Protokoll, und UDP, ein verbindungsloses Protokoll, die auf dem verbindungslosen Vermittlungsprotokoll IP der Vermittlungsschicht basieren.

Zwei CMIP-Anwendungsinstanzen können nur dann Management-Informationen miteinander austauschen, wenn sie eine Verbindung aufbauen und sie während des Informationsaustausches aufrechterhalten. Der Grund hierfür ist, daß CMIP ein verbindungsorientiertes Transportprotokoll benutzt, das auch zuverlässiges Transportprotokoll *(reliable transport protocol)* genannt wird. CMIP ist zuverlässig, weil eine Bestätigung der gesandten Nachrichten nach einem innerhalb der Transportschicht festgelegten Verfahren erwartet wird. Dabei bestimmt das Verfahren, nach welcher Zeitspanne und wie oft erneut zu senden ist [CFSD90b].
Im Gegensatz dazu benötigen zwei SNMP-Anwendungsinstanzen keinen Verbindungsaufbau vor dem Informationsaustausch und die Datenpakete können sofort geschickt werden. Der Grund hierfür besteht darin, daß SNMP ein verbindungsloses Transportprotokoll, auch unzuverlässiges Transportprotokoll *(unreliable transport protocol)* genannt, benutzt. Dabei soll unter einem unzuverlässigen Transportprotokoll ein Transportprotokoll verstanden werden, daß keine Garantie für die Ankunft der gesandten Nachrichten gewährleistet. In diesem Fall wird eine Sicherheit für die Ankunft der Nachrichten durch nochmaliges Übertragen erreicht, wobei die Anwendungen selbst und nicht die Transportschicht das erneute Senden der Nachrichten steuern.

Zum besseren Verständnis des Begriffs Zuverlässigkeit bezüglich der Transportdienste und -protokolle können als Analogien das Telefongespräch für einen verbindungsorientierten und der Brief für einen verbindungslosen Transportdienst genannt werden. Für ein Telefongespräch muß eine Verbindung zwischen den Gesprächspartnern vor dem Gespräch aufgebaut werden, während ein Brief einfach geschickt wird. Darüber hinaus benötigt der Brief im Gegensatz zu einem Telefongespräch weniger Ressourcen und technischen Aufwand.

Die Entwickler von SNMP haben das verbindungslose Protokoll UDP gewählt, um u.a. eine zusätzliche Belastung *(overhead)* zu vermeiden, die durch ein verbindungsorientiertes Protokoll TCP entsteht. SNMP spart durch diese Wahl Prozessorzeit, die bei jedem Datenaustausch für Verbindungsaufbau und -abbau benötigt wird. Demzufolge kann ein Agent auch schneller antworten, d.h. die Antwortzeit ist in diesem Fall kürzer. Im SNMP kann der Manager selbst die Kommunikationsparameter neu setzen und bestimmen, wie oft eine Nachricht geschickt werden soll, wenn eine Nachricht nicht empfangen worden ist. Im CMIP dagegen sind die Einstellung der Kommunikationsparameter und die Bestimmung der Anzahl von Sendeversuchen durch den Manager nicht möglich, da das verbindungsorientierte Protokoll diesbezüglich für Transparenz sorgt. Ein anderer wichtiger Vorteil des verbindungslosen Protokolls SNMP ist, daß die Meldungen trotz einer defekten Verbindung mit hoher Fehlerrate dem Manager geschickt werden können, was die Robustheit eines Netzwerkmanagements anzeigt.

Die OSI behandelt das OSI-Netzwerkmanagement als eine Anwendung wie jede andere OSI-Anwendung auf der Anwendungsschicht. So stehen CMIP und anderen Anwendungen, wie z.B. Dateiübertragung *(file transfer)* und elektronische Post *(electronic mail)*, verbindungsorientierte Transport- und Vermittlungsdienste zur Verfügung. Der Vorteil hierbei ist, daß der Empfang von Nachrichten am Empfangsort sichergestellt wird. Der Nachteil ist die Abhängigkeit von Diensten der höheren Schichten, die selbst im Falle eines Fehlers ihre Funktionsfähigkeit verlieren. Tritt ein Fehler auf, der einen Verbindungsaufbau verhindert, so können z.B. keine Meldungen dem Manager geschickt werden. Das Netzwerkmanagement soll aber gerade in solchen kritischen Zeiten, wenn die Funktionsfähigkeit des Netzes beeinträchtigt ist, die Möglichkeit der Benachrichtigung des Managers zur Verfügung stellen. Daher ist die Behandlung des OSI-Netzwerkmanagements als eine normale Anwendung als Nachteil anzusehen.

Das Protokoll UDP im SNMP-Netzwerkmanagement benutzt den verbindungslosen Vermittlungsdienst der Internetzschicht, während das Protokoll TP4 sowohl vom verbindungslosen als auch vom verbindungsorientierten Vermittlungsdienst unterstützt werden kann. Die Vorteile und Nachteile von verbindungsorientiertem und verbindungslosem Vermittlungsdienst sind eng miteinander verbunden:

- Im verbindungsorientierten Vermittlungsdienst liegt die Route für die Datenpakete fest. Dadurch wird weniger Verarbeitungszeit für ihre Weiterleitung benötigt. Im verbindungslosen Vermittlungsdienst muß jedes Zwischensystem die nächste Teilstrecke innerhalb der Gesamtroute berechnen, was zusätzliche Rechenzeit in Anspruch nimmt;

- Die Ressourcen werden im verbindungsorientierten Vermittlungsdienst reserviert. Dadurch sind die Endsysteme von der Verkehrslast relativ unabhängig. Im verbindungslosen Vermittlungsdienst dagegen können z.B. die Ressourcen so belastet werden, daß eine Verstopfung *(congestion)* im Netzwerk auftritt.

- Die Abrechnungmechanismen sind im verbindungsorientierten Vermittlungsdienst im Gegensatz zum verbindungslosen Vermittlungsdienst einfach zu realisieren, da eine Verbindung für den Datenaustausch aufgebaut wird.

- Der verbindungsorientierte Vermittlungsdienst ist langsam und teuer, da auch für kleine Datenmengen die Prozedur Verbindungsaufbau, Datenübertragung und Verbindungsabbau mit den zugehörigen Verhandlungen stattfinden soll, was im verbindungslosen Vermittlungsdienst nicht erforderlich ist.

- Im Falle des Ausfalls einer der die Verbindung unterstützenden Ressourcen wird der Datenaustausch unterbrochen, obwohl noch andere mögliche Wege zwischen Endsystemen im Netzwerk existieren. Der verbindungslose Vermittlungsdienst erweist sich diesbezüglich als robust.

- Die einer Verbindung zugeordneten Ressourcen sind für andere Verbindungen auch dann nicht verfügbar, wenn die Verbindung nicht benutzt wird. Im verbindungslosen Vermittlungsdienst können die Ressourcen effektiver benutzt werden.

### 4.5.6 Standards und Tests

Das Management-Protokoll CMIP im OSI-Netzwerkmanagement ist durch ISO/IEC normiert worden und stellt wie jedes andere ISO-Protokoll einen internationalen Standard dar. Im Rahmen der ISO-Standardisierung können die Herstellerimplementierungen auf die Konformität mit den Standards geprüft werden, indem sie den durch COS *(Corporation for Open Systems)* gelieferten Konformitätstests unterzogen werden (Abb. 38). Unter Konformität versteht man dann die Überprüfung auf die Einhaltung der Standards, wie z.B. das Vorhandensein von Protokollparametern, den Wertebereich von Protokollparametern oder die korrekte Folge der Übertragung von PDUs. Deshalb liefern die Konformitätstests keine Aussage über Interoperabilität zwischen Produkten oder Leistungsfähigkeit, Robustheit und Zuverlässigkeit der Produkte. So gibt ein Konformitätstest keine vollständige Garantie für die Zusammenarbeit unterschiedlicher Systeme, obwohl dadurch jedoch die Wahrscheinlichkeit einer erfolgreichen Zusammenarbeit *(interworking)* erhöht wird. Zusätzlich ermöglichen die Interop-Ausstellungen oder die anderen privaten Ausstellungen die Interoperabilität zwischen den Produkten verschiedener Hersteller zu testen und zu demonstrieren.

Das Management-Protokoll SNMP ist durch IAB *(Internet Activities Board)*, wie die anderen Protokolle des Internet-Protokollsatzes, normiert worden und stellt deshalb keinen offiziellen internationalen Standard dar. SNMP gilt im Gegensatz zum CMIP, welches ein De-jure-Standard ist, als ein De-facto-Standard [Buer90].
Die Vorgehensweise von IAB zur Entwicklung von Standards unterscheidet sich von der ISO-Strategie in der Art, daß die der IAB zur Standardisierung eingereichten Protokolle schon implementiert sein und verschiedene Interoperabilitätstests bestanden haben müssen, bevor sie als Standard gebilligt werden dürfen. Während die Entwicklung der Standards in IAB mehr auf den Aspekt der Funktionalität der Protokolle in der Praxis abzielt, ist der ISO-Standardisierungsprozeß vor allem damit beschäftigt, eine Übereinstimmung zwischen den Mitgliedern zu finden. So ist der ISO-Standardisierungsprozeß von den unterschiedlichen Interessen ihrer Mitglieder abhängig und kennzeichnet deshalb einen langwierigen Prozeß, welcher noch nicht abgeschlossen ist und einen direkten Einfluß auf die Verzögerung der Implementierung der Protokolle in Produkten hat [Jack91b].

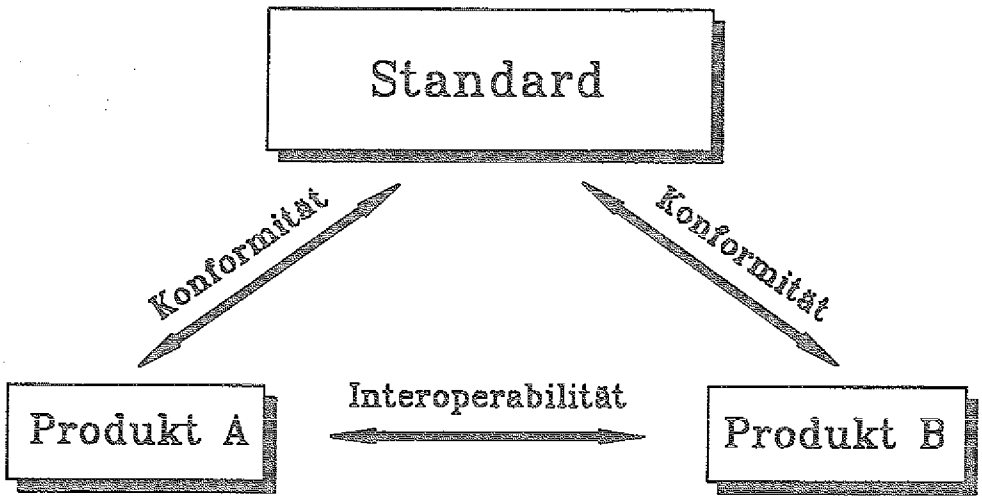



**Abb. 38. Konformität und Interoperabilität:** Die Konformitätstests erlauben die Produkte auf Einhaltung der Standards zu prüfen, obwohl sie keine Garantie für die Interoperabilität der zwei konformen Produkte geben.

Das OSI-Netzwerkmanagement-Forum (OSI/NM-Forum) und das GOSIP *(Government OSI Profile)* spielen eine wichtige Rolle bei der Förderung der OSI-Implementierung [Bona91]. Die OSI/NM-Forum setzt sich aus einer Gruppe von Herstellern zusammen, wie z.B. AT&T, Amdahl, British Telecom, Hewlett Packard usw., die im wesentlichen das Ziel verfolgen, die Entwicklung von OSI-basierten Management-Produkten zu beschleunigen. Darüber hinaus gehören z.B. die Durchführung der Konformitätstests in Kooperation mit COS und SPAG *(Standards Promotion and Application Group)* sowie die Definition von Umgebungen für die OSI-konformen Netzwerkmanagement-Anwendungen, wie z.B. Nachrichten-Formate oder Protokoll-Optionen, zu den Zielen dieser Gruppe [Carr90].
GOSIP kennzeichnet die Bemühungen der amerikanischen Regierung zur Durchsetzung der vorhandenen OSI-Protokolle, wo ihr Einsatz möglich ist [Fish90].

### 4.5.7 Verfügbarkeit der Produkte

Eine der wichtigsten Vorteile von SNMP besteht darin, daß es von vielen Produkten unterstützt wird. Die weite Verbreitung ist sicherlich in den minimalen Forderungen, die es an ein zu verwaltendes Netzwerkgerät stellt, zu begründen. Die Erweiterbarkeit der MIB beispielsweise ermöglicht, neue SNMP-unterstützende Geräte zu definieren, die der Anpassung zu den neuen oder zukünftigen Anforderungen dienen. Die Anzahl der SNMP-unterstützenden Hersteller wächst ständig. Darunter können beispielsweise Hersteller von Rechnern (IBM, DEC, HP, SUN), von LAN-Software (Novell, 3Com, IBM), von Internetz-Umgebung (Cisco, Hughes, Proteon, Wellfeet) und von TCP/IP-Software (Spider, Wollongong), sowie einige Hersteller von LAN-Hardware (SEL, Synoptics) genannt werden. Beispiele für die SNMP-unterstützenden Netzwerkmanagement-Produkte sind PSI-SNMP, SUN-SNM, HP-OpenView, IBM-AIX/NetView/6000 und DEC/EMA-DECmcc [Korz91, Scot90a, Jand91b].

CMIP stellt einen internationalen Standard dar und wird deshalb von vielen Anwendern bei der Beschaffung ihrer Netzwerkmanagement-Produkte verlangt. Die OSI-Produkte sind aber nicht verfügbar, wobei unter anderem die Nicht-Verfügbarkeit der Netze, die den OSI-Protokollsatz implementiert haben, die Ursache dafür sind. Die Hybridlösungen wie z.B. CMOT oder CMOL sowie auch die Bemühungen des OSI/NM-Forums haben das Ziel, den Übergang zum OSI-Netzwerkmanagement zu erleichtern, indem sie auf CMIP-basierende Produkte entwickeln, obwohl sie keine auf den Protokollen des OSI-Referenzmodells basierenden OSI-Netzwerkmanagement-Produkte repräsentieren [Carr90].

Im Zusammenhang mit der Verfügbarkeit der Produkte muß auch die Lösung DME *(Distributed Management Environment)* von OSF *(Open Software Foundation)* erwähnt werden. OSF ist eine Initiative vieler Hersteller, die das Ziel haben, Software bereitzustellen, die eine offene Systemumgebung sowie herstellerübergreifende verteilte Verarbeitung ermöglicht. Im Rahmen des Netzwerkmanagements hat OSF am 17. September 1991 ihre Auswahl für DME bekannt gegeben. Demnach ist DME die Integration der Technologien verschiedener Hersteller, um eine einheitliche Umgebung zur Verwaltung von Systemen, Netzwerken und Anwendungen zu schaffen, näher gekommen. DME ist so festgelegt worden, daß sie die Protokolle CMIP und SNMP sowie die entsprechenden Informationsmodellierungen (SMI), d.h. OSI-SMI und Internet-SMI unterstützt. So bietet DME als eine Mischung von Management-Komponenten verschiedener Hersteller eine wichtige Lösung für die Zukunft, deren Erfolg natürlich vom Verhalten der Hersteller abhängig ist [Heig91, Jack91d, Jack91c].

# 5 Zusammenfassung

In dieser Arbeit wird die Rolle des SNMP-Netzwerkmanagements als eine Zwischenlösung bis zur Verfügbarkeit von OSI-Lösungen untersucht. Dabei basiert die Untersuchung auf den bis Mitte 1991 verfügbaren Standards, die den Rahmen für die Beschreibung der einzelnen Netzwerkmanagement-Ansätze in dieser Arbeit vorgeben.

Zunächst wird auf das OSI-Netzwerkmanagement, das als ein De-jure-Standard gilt, eingegangen. So wird das Management-Protokoll CMIP zusammen mit den wichtigsten Mechanismen und Funktionalitäten dieser Architektur beschrieben, welche das OSI-Netzwerkmanagement deutlich als ein umfassendes Netzwerkmanagement kennzeichnen. Die OSI-Netzwerkmanagement-Standards enthalten aber keine Normung für die Netzwerkmanagement-Anwendungen, die beispielsweise die Überwachung relevanter Objekte initiieren, die Entscheidungen in Fehlersituationen treffen oder Schnittstellen zum Administrator bereitstellen. Dennoch ist das OSI-Netzwerkmanagement, insgesamt gesehen, sehr komplex.

Die Architektur des SNMP-Netzwerkmanagements wird unter zwei Aspekten betrachtet. Erstens wird das Ziel verfolgt, durch das Modell die zentral orientierte Management-Lösung des SNMP-Netzwerkmanagements zu betonen. Zu diesem Zweck werden die Komponenten vom SNMP-Netzwerkmanagement und ihre Wechselwirkungen zur Erfüllung der Management-Aufgabe dargestellt. Zweitens wird durch das SNMP-Netzwerkmanagement-Rahmenwerk eine detaillierte Beschreibung der das Rahmenwerk zusammensetzenden Komponenten, nämlich SMI, MIB und SNMP, gegeben. Die Betrachtung des Rahmenwerks hebt die Einfachheit des SNMP-Netzwerkmanagements im Gegensatz zur Komplexität des OSI-Netzwerkmanagements besonders deutlich hervor.
Im Gegensatz hierzu stellt das OSI-Netzwerkmanagement ein umfangreiches, komplexes Netzwerkmanagement dar, wobei die Komplexität auf die Globalität der Architektur und besonders auf die Vorgehensweise der ISO zur Lösung der Management-Aufgabe zurückzuführen ist.

Auf der Grundlage dieser Untersuchungen werden die beiden Netzwerkmanagement-Ansätze theoretisch auf die wesentlichen Ähnlichkeiten und Unterschiede geprüft. Der Vergleich zeigt, daß die Unterschiede zwischen den beiden Netzwerkmanagement-Ansätzen mehr von der Art und Weise ihrer Annäherung zur Lösung der Management-Aufgabe resultieren, als dies nur eine rein technische Frage zwischen den beiden Architekturen wäre. Während das SNMP-Netzwerkmanagement auf die Praxis zugeschnitten ist und die Einfachheit des Gesamtkonzepts in allen Überlegungen berücksichtigt wird, basiert das OSI-Netzwerkmanagement auf der Globalität und dem einheitlichen Informationsaustausch über Ressourcen und Objekte.
Das OSI-Netzwerkmanagement verfügt über das Management-Protokoll CMIP, welches der Kommunikation zwischen den Management-Prozessen in verschiedenen Systemen dient; SNMP ist das Management-Protokoll im SNMP-Netzwerkmanagement und wird zum Informationsaustausch zwischen der NMS und den Agenten benutzt. SNMP ist im Gegensatz zum CMIP, das den Management-Anwendungen eine objektorientierte Schnittstelle bereitstellt, attributorientiert.

Das SNMP-Netzwerkmanagement wird durch die minimalen, an ein Netzwerkgerät gestellten Forderungen, d.h. geringe Anforderung an Speicher- und Rechenkapazität, gekennzeichnet, was im OSI-Netzwerkmanagement nicht der Fall ist.
Obwohl das OSI-Netzwerkmanagement theoretisch ein mächtiges Werkzeug zur Steuerung und Überwachung der Netzwerke darstellt, enthält es einige Schwachpunkte, die ein Hindernis für die Nützlichkeit eines Netzwerkmanagements repräsentieren, wie z.B. die Zuordnung des Netzwerkmanagements zur Anwendungsschicht oder die Benutzung des verbindungsorientierten Transportmechanismus. Das Netzwerkmanagement muß auch dann funktionieren, wenn die anderen Anwendungen ausgefallen sind, was mittels eines verbindungslosen Transportmechanismus, wie im SNMP-Netzwerkmanagement, eher möglich ist als mit einem verbindungsorientierten Transportmechanismus, wo jeder Informationsaustausch mit einem Verbindungsaufbau und -abbau verbunden ist. Darüber hinaus zeigt sich das SNMP-Netzwerkmanagement sogar geeigneter als das OSI-Netzwerkmanagement, wenn es um Management nichtkomplexer Netzwerke und einfacher Netzwerkelemente geht.

Obwohl das SNMP-Netzwerkmanagement keinen ISO-Standard darstellt, spielt es wegen seiner spezifischen Merkmale eine bedeutende Rolle in der Lösung der heutigen Netzwerkmanagement-Aufgaben. So ist das SNMP-Netzwerkmanagement als eine gute Zwischenlösung auf dem Migrationsweg zu den OSI-Lösungen zu sehen. Tatsächlich wird aber in der Zukunft nur die Kooperation von verschiedenen Protokollen die wachsenden Netzwerkmanagement-Aufgaben bewältigen können.

# 6 Literaturverzeichnis


[BeCW90] Amatzia Ben-Artzi, Asheem Chandna, Unni Warrier. Network management of TCP/IP Networks: present and future. In: *IEEE Network Magazine, vol.4, no.4, p.35.* July, 1990.

[Benn89] Geoff Bennett. Simple solutions to managing networks. In: *Communications, vol.6, no.11, p.16.* November, 1989.

[Benn90] Geoff Bennett. The simple network management protocol. In: *Telecommunications, vol.24, no.2, p.21, International Edition.* February, 1990.

[Blac91] Uyless Black. *OSI- A Model for Computer Communications Standards.* Prentice Hall, Englwood Cliffs, New Jersey 07632, 1991. ISBN 0-13-637133-7.

[Bona90a] Christine Bonafield. Pointed competition among protocols. In: *Communications Week, issue n317, p.WP10.* September 10, 1990.

[Bona90b] Christine Bonafield. An imperfect god. In: *Communications Week, issue n317, p.WP6.* September 10, 1990.

[Bona91] Christine Bonafield. OSI: feeling the pull of technology. In: *Communications Week, issue n336, p.SR3.* January 28, 1991.

[Bors90] Paulina Borsook. Meditations on TCP/IP management. In: *Network World, issue n21, vol.v7, p.60.* May 21, 1990.

[Buer90] David J. Buerger. SNMP management will dominate in near future. In: *Info World, issue n29, vol.v12, p.29.* July 16, 1990.

[CaMa91] Jeffrey D. Case, Martin Marietta. Network Management and the Simple Network Management Protocol (SNMP). In: *Tutorial Proceedings for the 2nd International Symposium on Integrated Network Management.* Crystal City, Washington D.C., U.S.A., April 1991.

[Cari91] Smoot Carl-Mitchell. SNMP Query Language. In: *Datapro International Network Managment Service, Issue 1: Standards, Protocols, and Architectures.* McGraw-Hill Inc., Datapro Information Service Group, Delran, New Jersey 08075, 1991.

[Carr90] Jim Carr. Promises, promises: but OSI management products are scarce. In: *LAN Magazine, issue n6, vol.v5, p.113.* June 1990.

[Cerf88] Vinton G. Cerf. *IAB Recommendation for the Development of Internet Network Management Standards.* Request for Comments 1052. DDN Network Information Center. SRI International. April, 1988.

[Cerf89] Vinton G. Cerf. *Report of the Second Ad Hoc Network Management Review Group.* Request for Comments 1109. DDN Network Information Center. SRI International. August, 1989.

[CFSD89] Jeffrey D. Case, Mark S. Fedor, Martin L. Schoffstall, James R. Davin. Internet Network Management Using The Simple Network Management Protocol. In: *Proceedings 14th Conference on Local Computer Networks, Mineapolis, MN, USA, 10-12 Oct, 1989, p.156.* IEEE Compu. Soc. Press, 1989. ISBN 0-8186-1968-6.

[CFSD90a] Jeffrey D. Case, Mark S. Fedor, Martin L. Schoffstall, James R. Davin. *A Simple Network Management Protocol.* Request for Comments 1157, DDN Network Information Center, SRI International, May, 1990.

[CFSD90b] Jeffrey D. Case, Mark S. Fedor, Martin L. Schoffstall, James R. Davin. Keeping it simple. In: *Unix Review, vol.8, no.3, p.60.* March 1990.

[Come88] Douglas Comer. *Internetworking with TCP/IP: Principles, Protocols, and Architecture.* Prentice Hall, Englewood Cliffs, New Jersey 07632, 1988. ISBN 0-13-470154-2.

[Conr89] Dieter Conrads. *Datenkommunikation: Verfahren - Netze - Dienste.* Braunschweig, Wiesbaden: Vieweg, 1989. ISBN 3-528-04589-2.

[CrWa91] Diana Cripps, Jane Walters. OSI-Based Network Management. In: *Datapro International Network Managment Service, Issue 1: Standards, Protocols, and Architecture.* McGraw-Hill Inc., Datapro Information Service Group, Delran, New Jersey 08075, 1991.

[Davi88] John Davidson. *An Introduction to TCP/IP.* Springer-Verlag, New York Inc., 1988. ISBN 0-387-96651-X.

[DCFS87] James R. Davin, Jeffrey D. Case, Mark S. Fedor, Martin L. Schoffstall. *A Simple Gateway Monitoring Protocol.* Request for Comments 1028, DDN Network Information Center, SRI International, November, 1987.

[Doug91] Elizabeth Dougherty. CMOL simmers. In: *LAN Magazine, issue n3, vol.v6, p.77.* March, 1991.

[FeRS90] Mark S. Fedor, Michael S. Richards, Martin L. Schoffstall. Cutting management tasks down to size with SNMP. In: *LAN Techology, p.32.* March, 1990.

[Fish90] Sharon Fisher. The latest GOSIP. In: *Byte, p.212.* June, 1990.

[Fish91] Sharon Fisher. Duelling protocols. In: *Byte, vol.16, no.3, p.183.* March, 1991.

[GoSp90] Walter Gora, Reinhard Speyerer. *ASN.1 - Abstract Syntax Notation One*. Datacom Verlag, zweite Auflage, 1990. ISBN 3-89238-023-6.

[Heig91] Johannes Heigert. OSF/DME: kein klarer Gewinner. In: *Datacom, p.86*. November 1991.

[Herm91] James Herman. The network management protocol managerie. In: *Communications Week, issue n353, p.IG8*. May 27, 1991.

[Howe90] Charles L. Howe. Newsmaker - Pied Piper of SNMP tells how that TCP/IP protocol was born. In: *Data Communications International, p.59*. January, 1990.

[Hunt91] Jill Huntington-Lee. Network Management Functions. In: *Datapro International Network Managment Service, Issue 9: Issues and Strategies*. McGraw-Hill Inc., Datapro Information Service Group, Delran, New Jersey 08075, 1991.

[IAB91] Internet Activities Board. *IAB Official Protocol Standards*. Request for Comments 1250, DDN Network Information Center, SRI International, August, 1991.

[ISO7498] Information Processing Systems - Open Systems Interconnection - Basic Reference Model. International Organization for Standardization. International Standard 7498.

[ISO7498-4] Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management Framework. International Organization for Standardization and International Electrotechnical Committee. International Standard 7498-4.

[ISO8073AD1] Information Processing Systems - Open Systems Interconnection - Connection Oriented Transport Protocol Specification - Addendum 1: Network Connection Management Subprotocol. International Organization for Standardization and International Electrotechnical Committee. International Standard 8073 Addendum 1.

[ISO8571] Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management. International Organization for Standardization. International Standard 8571.

[ISO8649] Information Processing Systems - Open Systems Interconnection - Service definition for the Association Control Service Element. International Organization for Standardization. International Standard 8649.

[ISO8822] Information Processing Systems - Open Systems Interconnection - Connection oriented presentation service definition. International Organization for Standardization. International Standard 8822.

[ISO8824] Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1). International Organization for Standardization. International Standard 8824.

[ISO8825] Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). International Organization for Standardization. International Standard 8825.

[ISO8650] Information Processing Systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element. International Organization for Standardization. International Standard 8650.

[ISO9072-1] Information Processing Systems - Text Communication - Remote Operations - Part 1: Model, Notation and Service Definition. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 9072-1.

[ISO9072-2] Information Processing Systems - Text Communication - Remote Operations - Part 2: Protocol Specification. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 9072-2.

[ISO9595] Information Technology - Open Systems Interconnection - Common Management Information Service Definition. International Organization for Standardization and International Electrotechnical Committee. International Standard 9595.

[ISO9596] Information Technology - Open Systems Interconnection - Common Management Information Protocol Specification. International Organization for Standardization and International Electrotechnical Committee. International Standard 9596.

[ISO10026-1] Information Processing Systems - Open Systems Interconnection - Distributed Transaction Processing, Part 1: Model. International Organization for Standardization and International Electrotechnical Committee. Draft Proposal 10026-1.

[ISO10040] Information Technology - Open Systems Interconnection - Systems Management Overview. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10040.

[ISO10164-1] Information Technology - Open Systems Interconnection - Systems Management - Part 1: Object Management Function. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10164-1.

[ISO10164-2] Information Technology - Open Systems Interconnection - Systems Management - Part 2: State Management Function. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10164-2.

[ISO10164-3] Information Technology - Open Systems Interconnection - Systems Management - Part 3: Attributes for Representing Relationships. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10164-3.

[ISO10164-4] Information Technology - Open Systems Interconnection - Systems Management - Part 4: Alarm Reporting Function. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10164-4.

[ISO10164-5] Information Technology - Open Systems Interconnection - Systems Management - Part 5: Event Report Management Function. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10164-5.

[ISO10164-6] Information Technology - Open Systems Interconnection - Systems Management - Part 6: Log Control Function. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10164-6.

[ISO10164-7] Information Technology - Open Systems Interconnection - Systems Management - Part 7: Security Alarm Reporting Function. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10164-7.

[ISO10164-8] Information Technology - Open Systems Interconnection - Systems Management - Part 8: Security Audit Trail Function. International Organization for Standardization and International Electrotechnical Committee. Committee Draft 10164-8.

[ISO10164-9] Information Technology - Open Systems Interconnection - Systems Management - Part 9: Objects and Attributes for Access Control. International Organization for Standardization and International Electrotechnical Committee. Committee Draft 10164-9.

[ISO10164-10] Information Technology - Open Systems Interconnection - Systems Management - Part 10: Accounting Meter Function. International Organization for Standardization and International Electrotechnical Committee. Committee Draft 10164-10.

[ISO10164-11] Information Technology - Open Systems Interconnection - Systems Management - Part 11: Workload Monitoring Function. International Organization for Standardization and International Electrotechnical Committee. Committee Draft 10164-11.

[ISO10164-12] Information Technology - Open Systems Interconnection - Systems Management - Part 12: Test Management Function. International Organization for Standardization and International Electrotechnical Committee. Committee Draft 10164-12.

[ISO10164-13] Information Technology - Open Systems Interconnection - Systems Management - Part 13: Measurement Summarisation Function. International Organization for Standardization and International Electrotechnical Committee. Committee Draft 10164-13.

[ISO10165-1] Information Technology - Open Systems Interconnection - Management Information Services - Structure of Management Information - Part 1: Management Information Model. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10165-1.

[ISO10165-2] Information Technology - Open Systems Interconnection - Management Information Services - Structure of Management Information - Part 2: Definition of Management Information. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10165-2.

[ISO10165-4] Information Technology - Open Systems Interconnection - Management Information Services - Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objekts. International Organization for Standardization and International Electrotechnical Committee. Draft International Standard 10165-4.

[JaAg90] Bijendra N. Jain, Ashok K. Agrawala. *Open Systems Interconnection: Its Architecture and Protocols.* Elsevier Science Publishers B.V., 1990. ISBN 0-444-88490-4.

[Jack91a] Kelly Jackson. More uses for CMIP?. In: *Communications Week, issue n333, p.1.* January 7, 1991.

[Jack91b] Kelly Jackson. Which is the best route to take?. In: *Communications Week, issue n342, p.42.* March 11, 1991.

[Jack91c] Kelly Jackson. The holy war. In: *Communications Week, issue n342, p.39.* March 11, 1991.

[Jack91d] Kelly Jackson. Choosing the right environment. In: *Communications Week, issue n342, p.:40.* March 11, 1991.

[Jaco91] Ole J. Jacobsen. *A Glossary of Networking Terms.* Request for Comments 1208, NSF Network Service Center, March, 1991.

[Jand91a] Mary Jander. CMIP gets a new chance. In: *Data Communications, p.51.* September 1991.

[Jand91b] Mary Jander. Users rate SNMP multivendor network management systems. In: *Data Communications, p.113.* November 1991.

[Kauf90a] Franz-Joachim Kauffels. Netzwerk Management - Einführender Überblick. In: *Datacom - Netzwerk Management - Spezial.* Datacom Verlag, 1990.

[Kauf90b] Franz-Joachim Kauffels. *Netzwerk Management - Probleme, Standards, Strategien.* Datacom Verlag, 1990.

[Korz91] Paul Korzeniowski. Life in the LAN lane creates management jam. In: *Software Magazine, issue n9, vol.v11, p.66.* July, 1991.

[Kral90] Gray Krall. SNMP opens new lines of sight - with SNMP, ailing and failing devices on the internetwork are under management's watchful eye. In: *Data Communications, p.45.* March 21, 1990.

[Mard91] Jodi Mardesich. Users turn to SNMP to fullfill dreams of simplicity. In: *Info World, issue n2, vol.v13, p.S9.* January 14, 1991.

[McRo90] Keith McCloghrie, Marshall T. Rose. *Management Information Base Network Management of TCP/IP based internets.* Request for Comments 1156, DDN Network Information Center, SRI International, May, 1990.

[Paul90] Stephan Paulisch. Management gekoppelter, heterogener Lokaler Netze. In: *Datacom - Netzwerk Management - Spezial.* Datacom Verlag, 1990.

[Post80] Jonathan B. Postel. *User Datagram Protocol.* Request for Comments 768, DDN Network Information Center, SRI International, September, 1980.

[Post81a] Jonathan B. Postel. *Internet Protocol.* Request for Comments 791, DDN Network Information Center, SRI International, September, 1981.

[Post81b] Jonathan B. Postel. *Transmission Control Protocol.* Request for Comments 793, DDN Network Information Center, SRI International, September, 1981.

[Post82] Jonathan B. Postel. *Simple Mail Transfer Protocol.* Request for Comments 821, DDN Network Information Center, SRI International, August, 1982.

[Post83] Jonathan B. Postel. *TELNET Protocol Specification.* Request for Comments 854, DDN Network Information Center, SRI International, May, 1983.

[Post85] Jonathan B. Postel. *File Transfer Protocol.* Request for Comments 959, DDN Network Information Center, SRI International, October, 1985.

[Pres90] Randy Presuhn. Considering CMIP. In: *Data Communications, p.55.* March 21, 1990.

[Robe90] Erica Roberts. The simple approach to network management. In: *Communications, vol.7, no.11, p.63.* November, 1990.

[Robo90] Stefano Robotti. *Network Management Strategies.* Computer Technology Research Corp., 1990. ISBN 0-927695-55-3.

[RoMc90a] Marshall T. Rose, Keith McCloghrie. *Structure and Identification of Management Information for TCP/IP based internets.* Request for Comments 1155, DDN Network Information Center, SRI International, May, 1990.

[RoMc90b] Marshall T. Rose, Keith McCloghrie, James R. Davin. *Bulk Table Retrieval with the SNMP.* Request for Comments 1187, DDN Network Information Center, SRI International, October, 1990.

[Rose89] Ortwin Rose. Ein Überblick über die ISO/OSI-Management-Architektur. In: *Praxis der Informationsverarbeitung und Kommunikation, PIK 12(3), p.150.* Carl Hanser Verlag, München, 1989.

[Rose90a] Marshall T. Rose. *The Open Book: A Parctical Perspective on OSI.* Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1990. ISBN 0-13-643016-3.

[Rose90b] Ortwin Rose. OSI Management-Funktionen. In: *Praxis der Informationsverarbeitung und Kommunikation, PIK 13(4), p.191.* K. G. Saur Verlag, München, 1990.

[Rose90c] Marshall T. Rose. *Management Information Base Network Management of TCP/IP based internets: MIB-II.* Request for Comments 1158, DDN Network Information Center, SRI International, May, 1990.

[Rose91a] Marshall T. Rose. *The Simple Book: An Introduction to Management of TCP/IP-based Internets.* Prentice Hall, Englewood Cliffs, New Jersey 07632, 1991. ISBN 0-13-812611-9.

[Rose91b] Marshall T. Rose. Standards and Architecture - Network Management is Simple: You Just Need the "Right" Framework. In: *Proceedings of the IFIP TC6/WG6.6 Second International Symposium on Integrated Network Management.* North-Holland, IFIP, 1991. ISBN 0-444-89028-9.

[Rose91c] Marshall T. Rose. SNMP is the Way. In: *Communications Week, issue n356, p.17.* June 17,1991.

[Rose91d] Marshall T. Rose. *Management Information Base Network Management of TCP/IP based internets: MIB-II.* Request for Comments 1213, DDN Network Information Center, SRI International, March, 1991.

[Sabo91] L. Michael Sabo. An Overview of Simple Network Management Protocol. In: *Datapro International Network Managment Service, Issue 1: Standards, Protocols, and Architectures.* McGraw-Hill Inc., Datapro Information Service Group, Delran, New Jersey 08075, 1991.

[SaHu90] L. Michael Sabo, Jill Huntington-Lee. A heartbeat removed: SNMP demo reflects marketplace reality. In: *Communications Week, issue n321, p.S6.* October 8, 1990.

[Sant90] Michael Santifaller. *TCP/IP und NFS in Theorie und Praxis: UNIX in lokalen Netzen.* Addison-Wesley GmbH, Deutschland, 1990. ISBN 3-89319-246-8.

[ScCR91] Patricia Schnaidt, Jim Carr. Marshall Rose: principal scientist, performance systems int'l. In: *LAN Magazine, issue n3, vol.v6, p.26.* March, 1991.

[Scot90a] Karyl Scott. Taking care of business with SNMP. In: *Data Communications, p.31.* March 21, 1990.

[Scot90b] Karyl Scott. Brings order to chaos. In: *Data Communications, p.24.* March 21, 1990.

[Simp90] David Simpson. SNMP: simple but limited. In: *Systems Integration, vol.23, no.12, p.26.* December, 1990.

[Simp91] David Simpson. Solutions for network management. In: *Systems Integration, issue n3, vol.v24, p.34.* March, 1991.

[SITJ91] Chris Sluman, Jeremy Tucker, Tony Jeffree. OSI Management Standards, the Current Status of Work in ISO. In: *Tutorial Proceedings for the 2nd International Symposium on Integrated Network Management.* Crystal City, Washington D.C., U.S.A., April 1991.

[Stal89a] William Stallings. *Handbook of Computer-Communications Standards - The Open Systems Interconnection (OSI) Model and OSI-Related Standards - Volume 1.* Howard W. Sams & Company, Fifth Printing, 1989. ISBN 0-672-22664-2.

[Stal89b] William Stallings. *Handbook of Computer-Communications Standards - Local Network Standards Volume 2.* Howard W. Sams & Company, Fourth Printing, 1989. ISBN 0-672-22665-0.

[Step91] Peter Stephenson. Solving the SNMP manager puzzle. In: *LAN Magazine, issue n3, vol.v6, p.55.* March 1991.

[Tane90] Andrew S. Tanenbaum. *Computer-Netzwerke.* Wolfram's Fachverlag, 1990, 1. Auflage. ISBN 3-925328-079-3.

[VaCa90] Mark L. Van Name, Bill Catchings. Hard choices for network managers. In: *Byte, p.97.* November, 1990.

[Vand90] Chris VandenBerg. MIB-II extends SNMP interoperability. In: *Data Communications International, vol.19, no.13, p.97.* October, 1990.

[WBLH90] U. Warrier, L. Besaw. L. LaBarre, B. Handspicker. *The Common Management Information Services and Protocols for the Internet (CMOT and CMIP).* Request for Comments 1189. DDN Network Information Center. SRI International. October 1990.

[WiCa90] Bert Wijnen, Geoff Carpenter. An Introduction to TCP/IP Network Management. In: *Systems and Network Management, Technical Symposium.* La Hulpe, Belgium, October 1990.

[Yeon90] Wengyik Yeong. SNMP Query Language. In: *NYSERNet Technical Report 90-03-31-1.* March 31, 1990.

# Anhang A. Abkürzungen

| | |
|---|---|
| ACSE | Association Control Service Element |
| AD | ADdendum |
| AE | Application Entity |
| ANSI | American National Standards Institute |
| AP | Application Process |
| APDU | Application Protocol Data Unit |
| ARPA | Advanced Research Projects Agency |
| ASCII | American Standard Code for Information Interchange |
| ASE | Application Service Element |
| ASN.1 | Abstract Syntax Notation One |
| AT | Address Transmission |
| BER | Basic Encoding Rules |
| BSI | British Standards Institution |
| CD | Compact Disk |
| CLNP | ConnectionLess Network Protocol |
| CMIPDU | Common Management Information Protocol Data Unit |
| CMIPM | Common Management Information Protocol Machine |
| CMIS | Common Management Information Service |
| CMISE | Common Management Information Service Element |
| CMOL | CMIP over LLC |
| CMOT | CMIP over TCP/IP |
| COS | Corporation for Open Systems |
| CREN | Corporation for Research and Educational Networking |
| DAD | Draft ADdendum |
| DARPA | Defense Advanced Research Projects Agency |
| DIN | Deutsches Institut für Normung |
| DIS | Draft International Standard |
| DME | Distributed Management Environment |
| DoD | Department of Defense |
| DOS | Disk Operating System |
| DP | Draft Proposal |
| ECMA | European Computer Manufactures' Association |
| EGP | Exterior Gateway Protocol |
| EMA | Enterprise Management Architecture |
| ES-IS | End System to Intermediate System |
| FDDI | Fiber Distributed Data Interface |
| FSM | Finite State Machine |
| FTAM | File Transfer, Access and Management |
| FTP | File Transfer Protocol |
| GOSIP | Government OSI Profile |
| HLM | Heterogeneous LAN Management |
| IAB | Internet Activities Board |
| IANA | Internet Assigned Nummbers Authority |
| ICMP | Internet Control Message Protocol |
| IEC | International Electrotechnical Commission |

| | |
|---|---|
| **IEEE** | Institute of Electrical and Electronic Engineers |
| **IESG** | Internet Engineering Steering Group |
| **IETF** | Internet Engineering Task Force |
| **IP** | Internet Protocol |
| **IRSG** | Internet Research Steering Group |
| **IRTF** | Internet Research Task Force |
| **IS** | International Standard |
| **ISO** | International Standards Organization |
| **JTC** | Joint Technical Committee |
| **LAN** | Local Area Network |
| **LLC** | Logical Link Control |
| **LME** | Layer Management Entity |
| **LMI** | Layer Management Interface |
| **MAC** | Media Access Control |
| **MAPDU** | Management Application Protocol Data Unit |
| **MILNET** | MILitary NETwork |
| **MIB** | Management Information Base |
| **MIPS** | Million Instructions Per Second |
| **MIS** | Management Information System |
| **MIT** | Massachusetts Institute for Technology |
| **MOPS** | Management Operations Per Second |
| **NDIS** | Network Driver Interface Specification |
| **NMS** | Network Management Station |
| **NOC** | Network Operations Center |
| **NSD** | Network Statistics Database |
| **NSFNET** | National Science Foundation NETwork |
| **NVT** | Network Virtual Terminal |
| **NYSERNet** | New York State Education and Research Network |
| **ODI** | Open Data link Interface |
| **OSF** | Open Software Foundation |
| **OSI** | Open Systems Interconnection |
| **OSI-BRM** | Open Systems Interconnection - Basic Reference Modell |
| **OSIE** | Open Systems Interconnection Environment |
| **PC** | Personal Computer |
| **PCI** | Protocol Control Information |
| **PDAD** | Proposed Draft ADdendum |
| **PDU** | Protocol Data Unit |
| **PLP** | Packet Level Protocol |
| **QOS** | Quality Of Service |
| **RFC** | Request For Comments |
| **ROIVapdu** | ROse InVoke application protocol data unit |
| **RORSapdu** | ROse ReSult application protocol data unit |
| **ROSE** | Remote Operations Service Element |
| **SACF** | Single Association Control Function |
| **SAP** | Service Access Point |
| **SC** | Sub-Committee |
| **SGMP** | Simple Gateway Management Protocol |
| **SMAE** | Systems Management Application Entity |

| | |
|---|---|
| SMAP | Systems Management Application Process |
| SMASE | Systems Management Application Service Element |
| SMFA | Specific Management Functional Area |
| SMFU | Systems Management Functional Unit |
| SMI | Structure of Management Information |
| SMIP | Specific Management Information Protocol |
| SMIS | Specific Management Information Service |
| SMTP | Simple Mail Transfer Protocol |
| SNA | Systems Network Architecture |
| SNMP | Simple Network Management Protocol |
| SPAG | Standards Promotion and Application Group |
| TC | Technical Committee |
| TCP | Transmission Control Protocol |
| TP | Transaction Processing |
| UDP | User Datagram Protocol |
| WD | Working Document |
| WG | Working Group |

# Anhang B. Liste der in MIB-II definierten Objekte

## *Systemgruppe*

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 system 1 sysDescr 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 system 1 sysObjectID 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 system 1 sysUpTime 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 system 1 sysContact 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 system 1 sysName 5
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 system 1 sysLocation 6
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 system 1 sysServices 7
```

## *Schnittstellengruppe*

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifNumber 1

iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifIndex 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifDescr 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifType 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifMtu 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifSpeed 5
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifPhysAddress 6
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifAdminStatus 7
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifOperStatus 8
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifLastChange 9
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifInOctets 10
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifInUcastPkts 11
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifInNUcastPkts 12
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifInDiscards 13
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifInErrors 14
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifInUnknownProtos 15
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifOutOctets 16
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifOutUcastPkts 17
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifOutNUcastPkts 18
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifOutDiscards 19
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifOutErrors 20
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifOutQLen 21
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 interfaces 2 ifTable 2 ifEntry 1 ifSpecific 22
```

## *AT-Gruppe*

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 at 3 atTable 1 atEntry 1 atIfIndex 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 at 3 atTable 1 atEntry 1 atPhysAddress 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 at 3 atTable 1 atEntry 1 atNetAddress 3
```

## *IP-Gruppe*

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipForwarding 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipDefaultTTL 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipInReceives 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipInHdrErrors 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipInAddrErrors 5
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipForwDatagrams 6
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipInUnknownProts 7
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipInDiscards 8
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipInDelivers 9
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipOutRequests 10
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipOutDiscards 11
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipOutNoRoutes 12
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipReasmTimeout 13
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipReasmReqds 14
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipReasmOKs 15
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipReasmFails 16
```

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipFragOKs 17
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipFragFails 18
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipFragCreates 19

iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipAddrTable 20 ipAddrEntry 1 ipAdEntAddr 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipAddrTable 20 ipAddrEntry 1 ipAdEntIfIndex 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipAddrTable 20 ipAddrEntry 1 ipAdEntNetMask 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipAddrTable 20 ipAddrEntry 1 ipAdEntBcastAddr 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipAddrTable 20 ipAddrEntry 1 ipAdEntReasmMaxSize 5

iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteDest 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteIfIndex 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteMetric1 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteMetric2 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteMetric3 5
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteMetric4 6
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteNextHop 7
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteType 8
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteProto 9
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteAge 10
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipRoutingTable 21 ipRoute 1 ipRouteMask 11

iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipNetToMediaTable 22 ipNetToMediaEntry 1 ipNetToMediaIfIndex 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipNetToMediaTable 22 ipNetToMediaEntry 1 ipNetToMediaPhysAddress 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipNetToMediaTable 22 ipNetToMediaEntry 1 ipNetToMediaNetAddress 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 ip 4 ipNetToMediaTable 22 ipNetToMediaEntry 1 ipNetToMediaType 4
```

### *ICMP-Gruppe*

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInMsgs 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInErrors 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInDestUnreachs 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInTimeExcds 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInParmProbs 5
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInSrcQuenchs 6
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInRedirects 7
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInEchos 8
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInEchoReps 9
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInTimestamps 10
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInTimestampReps 11
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInAddrMasks 12
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpInAddrMaskReps 13
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutMsgs 14
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutErrors 15
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutDestUnreachs 16
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutTimeExcds 17
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutParmProbs 18
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutSrcQuenchs 19
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutRedirects 20
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutEchos 21
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutEchoReps 22
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutTimeStamps 23
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutTimeStampReps 24
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutAddrMasks 25
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 icmp 5 icmpOutAddrMaskReps 26
```

### *TCP-Gruppe*

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpRtoAlgorithm 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpRtoMin 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpRtoMax 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpMaxConn 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpActiveOpens 5
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpPassiveOpens 6
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpAttemptFails 7
```

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpEstabResets 8
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpCurrEstab 9
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpInSegs 10
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpOutSegs 11
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpRetransSegs 12

iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpConnTable 13 tcpConnEntry 1 tcpConnState 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpConnTable 13 tcpConnEntry 1 tcpConnLocalAddress 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpConnTable 13 tcpConnEntry 1 tcpConnLocalPort 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpConnTable 13 tcpConnEntry 1 tcpConnRemAddress 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpConnTable 13 tcpConnEntry 1 tcpConnRemPort 5

iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpInErrs 14
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 tcp 6 tcpOutRsts 15
```

### *UDP-Gruppe*

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 udp 7 udpInDatagrams 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 udp 7 udpNoPorts 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 udp 7 udpInErrors 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 udp 7 udpOutDatagrams 4

iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 udp 7 udpTable 5 udpEntry 1 udpLocalAddress 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 udp 7 udpTable 5 udpEntry 1 udpLocalPort 2
```

### *EPG-Gruppe*

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpInMsgs 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpInErrors 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpOutMsgs 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpOutErrors 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighState 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighAddr 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighAs 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighInMsgs 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighInErrs 5
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighOutMsgs 6
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighOutErrs 7
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighInErrMsgs 8
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighOutErrMsgs 9
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighStateUps 10
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighStateDowns 11
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighIntervalHello 12
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighIntervalPoll 13
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighMode 14
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighEventTrigger 15
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 egpNeighTable 5 egpNeighEntry 1 egpNeighInMsgs 4

iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 egp 8 epgAs 6
```

### *Übertragungsgruppe*

-- Platzhalter --

### *SNMP-Gruppe*

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInPkts 1
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutPkts 2
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInBadVersions 3
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInBadCommunityNames 4
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInBadCommunityUses 5
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInASNParseErrs 6
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInBadTypes 7
```

```
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInTooBigs 8
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInNoSuchNames 9
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInBadValues 10
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInReadOnlys 11
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInGenErrs 12
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInTotalReqVars 13
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInTotalSetVars 14
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInGetRequests 15
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInGetNexts 16
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInSetRequests 17
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInGetResponses 18
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpInTraps 19
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutTooBigs 20
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutNoSuchNames 21
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutBadValues 22
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutReadOnlys 23
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutGenErrs 24
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutGetRequests 25
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutGetNexts 26
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutSetRequests 27
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutGetResponses 28
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpOutTraps 29
iso 1 org 3 dod 6 internet 1 mgmt 2 mib 1 snmp 11 snmpEnableAuthTraps 30
```

# Anhang C. TOASTER-MIB

Zur Verdeutlichung der minimalen, vom SNMP an einen verwalteten Knoten gestellten Forderungen wurden zur Präsentation einige gewöhnliche Haushaltgeräte, wie z.B. Toaster, CD-Player und Spielzeuge durch SNMP gesteuert. Im folgenden wird die MIB des von John Romkey während der zwei letzten Interops repräsentierten Toasters angegeben. Die TOASTER-MIB[54] soll hier u.a. zum Verständnis der Syntax von Objekten mittels eines alltäglichen Geräts, nämlich verwalteten Toasters, beitragen.

```
        TOASTER-MIB DEFINITIONS ::= BEGIN

        IMPORTS enterprises
                    FROM RFC1155-SMI
                OBJECT-TYPE
                    FROM RFC-1212
                DisplayString
                    FROM RFC-1213;


epilogue        OBJECT IDENTIFIER ::= { enterprises 12 }
toaster         OBJECT IDENTIFIER ::= { epilogue 2 }

-- toaster MIB

toasterManufacturer     OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The name of the toaster's manufacturer. For instance,
                 Sunbeam."
        ::= { toaster 1 }

toasterModelNumber      OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The name of the toaster's model. For instance,
                 Radiant Automatic."
        ::=     { toaster 2 }

toasterControl  OBJECT-TYPE
        SYNTAX  integer {
                        up(1),
                        down(2)
                        }
```

---

54 Quelle: John Romkey, Epilogue Technology,
USENET/UUCP/Internet: romkey@asylum.sf.ca.us

```
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable controls the current state of the toaster.
                 To begin toasting, set it to down(2). To abort toasting
                 (perhaps in the event of an emergency), set it to
                 up(1)."
        ::=     { toaster 3 }

toasterDoneness OBJECT-TYPE
        SYNTAX  integer(1..10)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable controls how well done ensuing toast should be
                 on a scale of 1 to 10. Toast made at 10 is generally
                 considered unfit for human consumption; toast made at 1
                 is lightly warmed."
        ::=     { toaster 4 }

toasterToastType        OBJECT-TYPE
        SYNTAX  INTEGER {
                        white-bread(1),
                        wheat-bread(2),
                        wonder-bread(3),
                        frozen-waffle(4),
                        frozen-bagel(5),
                        hash-brown(6),
                        other(7)
                        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable informs the toaster of the type of material
                 being toasted. The toaster uses this information combined
                 with toasterToastDoneness to compute how long the
                 material must be toasted for to achieve the desired
                 doneness."
        ::=     { toaster 5 }

        END
```

# Anhang D. Gute Architektur?

Die folgende Erzählung[55] verdeutlicht auf einer unterhaltsamen Basis, was die SNMP-Befürworter unter einer guten Architektur und ihrer Vorgehensweise verstehen, und wie sie die Annäherung der CMIP-Entwickler zum Problem des Netzwerkmanagements einschätzen. In der Erzählung läßt ein König zwei von seinen Beratern für einen Test holen. Der erste Berater repräsentiert die Vorgehensweise der SNMP-Entwickler, und der zweite Berater stellt die Denkweise der OSI-Entwickler dar.

> Once upon a time, in a kingdom not far from here, a king summoned two of his advisors for a test. He showed them both a shiny metal box with two slots in the top, a control knob, and a lever. „What do you think this is?"
>
> One advisor, an engineer, answered first. „It is a toaster," he said. The king asked, „How would you design an embedded computer for it?" The engineer replied, „Using a four-bit microcontroller, I would write a simple program that reads the darkness knob and quantizes its position to one of 16 shades of darkness, from snow white to coal black. The program would use that darkness level as the index to a 16-element table of initial timer values. Then it would turn on the heating elements and start the timer with the initial value selected from the table. At the end of the time delay, it would turn off the heat and pop up the toast. Come back next week, and I'll show you a working prototype."
>
> The second advisor, a computer scientist, immediately recognized the danger of such short-sighted thinking. He said, „Toasters don't just turn bread into toast, they are also used to warm frozen waffles. What you see before you is really a breakfast food cooker. As the subjects of your kingdom become more sophisticated, they will demand more capabilities. They will need a breakfast food cooker that can also cook sausage, fry bacon, and make scrambled eggs. A toaster that only makes toast will soon be obsolete. If we don't look to the future, we will have to completely redesign the toaster in just a few years."
>
> „With this in mind, we can formulate a more intelligent solution to the problem. First, create a class of breakfast foods. Specialize this class into subclasses: grains, pork, and poultry. The specialization process should be repeated with grains divided into toast, muffins, pancakes, and waffles; pork divided into sausage, links and bacon; and poultry divided into scrambled eggs, hard-boiled eggs, poached eggs, fried eggs, and various omelet classes."
>
> „The ham and cheese omelet class is worth special attention because it must inherit characteristics from the pork, dairy, and poultry classes. Thus, we see that the problem cannot be properly solved without multiple inheritance. At run time, the program must create the proper object and send a message to the object that says, 'Cook yourself.' The semantics of this message depend, of course, on the kind of object, so they have a different meaning to a piece of toast than to scrambled eggs."
>
> „Reviewing the process so far, we see that the analysis phase has revealed that the primary requirement is to cook any kind of breakfast food. In the design phase, we have discovered some derived requirements. Specifically, we need an object-oriented language with multiple inheritance. Of course, users don't want the eggs to get cold while the bacon is frying, so concurrent processing is required, too."

---

55 Quelle: [Rose91b], Seite24-25.

„We must not forget the user interface. The lever that lowers the food lacks versatility, and the darkness knob is confusing. Users won't buy the product unless it has a user-friendly, graphical interface. When the breakfast cooker is plugged in, users should see a cowboy boot on the screen. Users click on it, and the message 'Booting UNIX v. 8.3' appears on the screen. (UNIX 8.3 should be out by the time the product gets to the market.) Users can pull down a menu and click on the foods they want to cook."

„Having made the wise decision of specifying the software first in the design phase, all that remains is to pick an adequate hardware platform for the implementation phase. An Intel 80386 with 8MB of memory, a 30MB hard disk, and a VGA monitor should be sufficient. If you select a multitasking, object oriented language that supports multiple inheritance and has a built-in GUI, writing the program will be a snap. (Imagine the difficulty we would have had if we had foolishly allowed a hardware-first design strategy to lock us into a four-bit microcontroller!)."

The king had the computer scientist thrown in the moat, and they all lived happily ever after.

Die vorliegende Arbeit wurde als Diplomarbeit am Lehrstuhl für Technische Informatik und Computerwissenschaften der Rheinisch-Westfälischen Technischen Hochschule Aachen erstellt.

Dem Lehrstuhlinhaber Herrn Prof. Dr. F. Hoßfeld danke ich für die Möglichkeit, die Arbeit am Zentralinstitut für Angewandte Mathematik des Forschungszentrums Jülich (KFA) anfertigen zu können.
Mein besonderer Dank gilt Herrn Dipl.-Inform. W. Elmenhorst für die Betreuung und konstruktiven Diskussionen.

FORSCHUNGSZENTRUM JÜLICH GmbH

KFA

Jül-2622
Mai 1992
ISSN 0366-0885