X.500 : eine Norm für Verzeichnisdienste

Autor(en): **Plattner, Bernhard / Lanz, Cuno / Zogg, Andreas**

Objekttyp: **Article**

Zeitschrift: **Technische Mitteilungen / Schweizerische Post-, Telefon- und Telegrafenbetriebe = Bulletin technique / Entreprise des postes, téléphones et télégraphes suisses = Bollettino tecnico / Azienda delle poste, dei telefoni e dei telegrafi svizzeri**

Band (Jahr): **68 (1990)**

Heft 3

PDF erstellt am: **22.07.2024**

Persistenter Link: https://doi.org/10.5169/seals-876195

X.500 – eine Norm für Verzeichnisdienste*

Bernhard PLATTNER, Cuno LANZ und Andreas ZOGG, Zürich

Zusammenfassung. *Elektronische Verzeichnisse sind die «Telefonbücher» moderner Kommunikationssysteme. Sie vereinfachen deren Nutzung, indem sie Informationen über Teilnehmer und beteiligte technische Systeme bereithalten und auf benützerfreundliche Art zugänglich machen. Die Autoren diskutieren die internationalen Normen, die im Hinblick auf eine weltweite Verknüpfung von Verzeichnisdiensten als Empfehlungen der CCITT unter der Bezeichnung X.500 verabschiedet wurden. Nach einer Übersicht werden die den Empfehlungen zugrunde liegenden Konzepte vorgestellt. Das Informationsmodell sowie die zur Verfügung stehenden Operationen und Aspekte der Verteilung der Daten auf verschiedene Rechner werden behandelt. Die Einbettung von X.500 in das OSI-Referenzmodell wird diskutiert und – als Beispiel – die Beziehung zu elektronischen Mitteilungssystemen konkretisiert. Zum Schluss wird das aktuelle Thema der Sicherheit kurz aufgegriffen und auf einige Probleme in X.500 und deren mögliche Lösungen eingegangen.*

X.500 – une norme pour les services d'annuaires

Résumé. *Les annuaires électroniques sont les «listes du téléphone» des systèmes de communication modernes. Ils en simplifient l'utilisation, vu qu'ils contiennent les informations concernant les abonnés et les systèmes techniques participants et qu'ils rendent ces informations accessibles de manière simple. Les auteurs discutent les normes internationales qui, dans l'optique d'une interconnexion des services d'annuaires au niveau mondial, ont été mises au point sous la forme des Recommandations du CCITT X.500. Après un aperçu général, on présente les conceptions à la base des recommandations. Le modèle d'information, les opérations à disposition et les aspects de la répartition des données sur plusieurs ordinateurs sont abordés. L'intégration des recommandations X.500 dans le modèle de référence OSI est discutée et les relations avec les systèmes de messagerie électronique sont concrétisées à la lumière d'un exemple. Pour terminer, le thème actuel de la sécurité est brièvement abordé et l'on soulève quelques problèmes relatifs à la norme X.500, tout en proposant des solutions possibles.*

X.500 – una norma per i servizi d'elenchi

Riassunto. *Gli elenchi elettronici sono gli «elenchi telefonici» dei moderni sistemi di comunicazione. Il loro uso risulta semplificato in quanto essi approntano le informazioni su utenti e corrispettivi sistemi, rendendole nel contempo facilmente accessibili. Gli autori discutono le norme internazionali che, in vista di un'interconnessione mondiale dei servizi d'elenchi, sono state emanate dal CCITT sotto forma di raccomandazioni con la designazione X.500. I concetti basati secondo dette raccomandazioni vengono qui esposti per sommi capi. Vengono in particolare trattati il modello d'informazione, le operazioni messe a disposizione e gli aspetti legati alla distribuzione dei dati sui differenti elaboratori. Viene discussa la collocazione della norma X.500 nell'ambito del modello di riferimento OSI e si rende concreta, a titolo d'esempio, la sua relazione con i sistemi di messaggeria elettronica. Per finire si tocca brevemente il tema, molto d'attualità, della sicurezza, con un accenno ad alcuni problemi connessi con la norma X.500 ed alle possibili soluzioni.*

1 Einführung

11 Eigenschaften von Verzeichnisdiensten

Heutige und vermehrt noch künftige Kommunikationssysteme sind komplexe Gebilde mit weltweiter Ausdehnung, die aus einer grossen Zahl von technischen Bestandteilen (Rechner, Prozesse, Dateisysteme, elektronische Briefkästen usw.) und nichttechnischen Einheiten (menschliche Teilnehmer, Organisationen) bestehen. Alle diese Komponenten sind *reale Objekte*. Ein Verzeichnis speichert Informationen über reale Objekte; ein reales Objekt wird somit durch die gespeicherte Information modelliert. Das Modell eines realen Objektes ist ein *logisches Objekt* oder kurz ein *Objekt*. Die zu einem Objekt gehörenden Daten werden als *Eintrag* im Verzeichnis gespeichert. Damit reale Objekte in verteilten Anwendungen eindeutig identifiziert werden können, werden sie mit Namen versehen. Ein reales Objekt kann durchaus mehrere Namen haben. Da Namen oft von menschlichen Benützern verwendet werden, müssen sie *benützerfreundlich* sein, das heisst, der Name eines realen Objektes sollte mit grosser Wahrscheinlichkeit aufgrund bekannter Eigenschaften desselben erraten werden können. Namen sollten zudem möglichst lange gültig sein.

Für die Erstellung von Kommunikationsbeziehungen zwischen realen Objekten werden üblicherweise jedoch *Adressen* benötigt. Diese bezeichnen die *Position eines realen Objektes* bezogen auf die Architektur des Systems. Da sich die Adresse eines Objektes aus der konkreten Ausführung eines Kommunikationssystems ableitet, ist sie oft für menschliche Benützer nicht geeignet. Die Adresse eines Benützers des OSI-Netzwerk-Dienstes beispielsweise ist eine Grösse mit bis zu 40 dezimalen Ziffern und erfüllt somit nicht das Kriterium der Benützerfreundlichkeit. Wenn die Netztopologie geändert wird, müssen oft einem Teil der realen Objekte neue Adressen zugeteilt werden. Adressen sind somit als Namen ungeeignet, obwohl auch sie reale Objekte eindeutig identifizieren.

Aus diesen einleitenden Bemerkungen folgt direkt die Hauptaufgabe von *Verzeichnisdiensten:* Sie ordnen dem Namen eines realen Objektes eine Menge von Werten zu. Diese Werte umfassen nicht nur die Adresse, sondern grundsätzlich alle wissenswerte Information in Form von Text, Sprache, Bildern usw. Mögliche Anwendungen sind die Speicherung von Passwörtern, die Verteilung von Chiffrierschlüsseln und die Verwaltung von Daten für die Verrechnung von Leistungen. Aus dieser Definition entstehen zusätzliche Anforderungen an Verzeichnisdienste. So stellt sich sofort die Frage, ob die gespeicherte Information, das sogenannte *Verzeichnis,* schützenswert ist und deshalb nur berechtigten Zugriffen durch autorisierte Teilnehmer ausgesetzt werden darf.

Der Verzeichnisdienst umfasst Operationen für die Abfrage und die Änderung des Verzeichnisses. Die *Abfrageoperationen* können grob in zwei Klassen aufgeteilt

* Nachdruck aus dem im Addison-Wesley Verlag (Deutschland) erschienenen Buch «Elektronische Post und Datenkommunikation» von Professor Bernhard Plattner, ETH Zürich, und Mitautoren. Mit freundlicher Genehmigung des Verlages.

werden: *White-Pages-Abfragen* liefern die zu einem oder mehreren gegebenen Namen gespeicherte Information, und *Yellow-Pages-Abfragen* liefern die Namen jener realen Objekte, die den in der Abfrage definierten Kriterien entsprechen. Die *Änderungsoperationen* umfassen das Einfügen und Entfernen von Einträgen oder Komponenten davon. Eine weitere Kategorie von Operationen ist für die *Verwaltung* des Verzeichnisses notwendig.

12 Anforderungen an Verzeichnissysteme

Als *Verzeichnissystem* wird eine Menge von realen Systemen im Sinn von OSI bezeichnet; es verwaltet das Verzeichnis und erbringt den Verzeichnisdienst. Mit der Bereitstellung rechnergestützter Kommunikationsdienste müssen auch Verzeichnisdienste für Rechneranwendungen zugänglich sein. Es gibt zwei offensichtliche Gründe dafür, dass Verzeichnissysteme verteilt sind. Einerseits sind moderne Kommunikationssysteme verteilte Anwendungen, die von einer grossen Anzahl von öffentlich-rechtlichen und privaten Organisationen betrieben werden, und anderseits erzeugt die Beschreibung der realen Objekte solcher weltweiter Systeme ein umfangreiches Datenvolumen.

Effiziente, verteilte Datenbanken sind heute noch Gegenstand der Forschung, das heisst, befriedigende Lösungen sind noch nicht bekannt. Während Verzeichnissysteme in einigen Aspekten als verteilte Datenbanken betrachtet werden können, erlauben sie jedoch Vereinfachungen, die sie einerseits als Forschungsgegenstand besonders interessant machen, anderseits auch eine effiziente Verwirklichung ermöglichen.

Verzeichnissysteme haben mit verteilten Datenbanken folgende Eigenschaften gemeinsam: Die Daten werden verteilt und aus Gründen der schnellen Zugreifbarkeit und hohen Verfügbarkeit repliziert. An die Verfügbarkeit werden in beiden Fällen hohe Erwartungen gestellt. In bezug auf die Konsistenz der gespeicherten Daten sind die Anforderungen jedoch unterschiedlich: Operationen auf verteilten Datenbanken müssen die Konsistenz der Daten garantieren; dagegen sollen solche auf Verzeichnissen *mit einer hohen Wahrscheinlichkeit* korrekte Daten liefern. Fehler sind tolerierbar, da Verzeichnissysteme viel häufiger abgefragt als geändert werden und die gelieferten Daten für Nachfolgeoperationen – etwa zum Erstellen einer Verbindung zu einem Kommunikationspartner mit einer erfragten Adresse – verwendet werden und fehlerhafte Angaben bei dieser Gelegenheit festgestellt werden können. Verzeichnissysteme dürfen also eine *beschränkte Konsistenz der Daten* aufweisen, die so definiert ist, dass nach Änderungen des Verzeichnissystems dieses längstens für ein gegebenes Zeitintervall inkonsistent sein darf.

Verzeichnissysteme gibt es schon seit längerer Zeit. Beispiele für in Betrieb befindliche Systeme sind:

- *The Domain Name System,* ein Verzeichnissystem für das sogenannte «ARPA Internet», das von der «Defence Advanced Research Projects Agency» (DARPA) unter Schirmherrschaft des US-Verteidigungsministeriums getragen wird und einen Zusammenschluss verschiedener Rechnernetze darstellt.
- *Grapevine,* ein verteiltes System, das im «Xerox Palo Alto Research Center» entwickelt wurde und heute bei der «Xerox Corporation» als unternehmensinternes System in Betrieb steht. «Grapevine» ist in erster Linie ein elektronisches Mitteilungssystem, welches eine sichere Übermittlung von Meldungen garantiert sowie Sender wie Empfänger von Meldungen authentifiziert. Daneben hat es die Funktion eines Verzeichnisdienstes. Die Umgebung von «Grapevine» bilden etwa 1500 Rechner auf etwas mehr als 50 lokalen Netzen, die zum sogenannten «Xerox Research Internet» zusammengekoppelt sind.
- *The Clearinghouse,* ein Verzeichnissystem, das aus der Verzeichnisdienstkomponente von «Grapevine» entstand. Viele seiner Konzepte sind aus Ideen weiterentwickelt, die im «Grapevine Registration Service» verwirklicht sind. Auch «The Clearinghouse» entstand bei Xerox.

13 Übersicht über die X.500-Empfehlungen

Verzeichnisdienste waren und sind Gegenstand der Normierung: Arbeitsgruppen der *International Standardisation Organisation* (ISO) in Zusammenarbeit mit der *International Electrotechnical Commission* (IEC) arbeiten gegenwärtig an der Weiterentwicklung des internationalen Standard ISO 9594, der in die acht Teile ISO 9594-1 bis ISO 9594-8 unterteilt ist. Das CCITT hat an seiner Plenarversammlung Ende 1988 Verzeichnisdienste unter der Bezeichnung *X.500* als Empfehlung verabschiedet.

Diese Normierungsarbeiten sind Teil der Normierung der Kommunikation offener Systeme (Open Systems Interconnection, OSI); der OSI-Verzeichnis-Dienst ist dabei als eine Sammlung von *Application Service Elements* (ASE) spezifiziert und somit in die gegenwärtig stark vorangetriebene Normierung der Anwendungsschicht integriert.

Die CCITT-Empfehlungen für Verzeichnisdienste enthalten in X.500 eine Übersicht über die Konzepte und das Modell eines standardisierten Verzeichnisdienstes und in X.501, X.520 und X.521 eine Beschreibung des Informationsmodells. Die abstrakte Dienstspezifikation ist in X.511 definiert und die Beschreibung der zu verwendenden Protokolle in X.519. X.518 diskutiert die Aspekte der Verteilung des Verzeichnisdienstes, und die Sicherheitsaspekte schliesslich sind in X.509 zu finden. Da die Normierung von Verzeichnisdiensten im Kontext der OSI-Normierung zu sehen ist, sind zusätzlich zahlreiche Bezüge auf neuere Normen der Anwendungsschicht notwendig, besonders auf jene der Serie X.200.

Die folgenden Erläuterungen und Aussagen dieses Kapitels sind alle im Kontext von X.500 zu verstehen. Die qualifizierenden Bezeichnungen «normenkonform», «standardisiert», «X.500-konform» usw. werden deshalb weggelassen.

2 Informationsmodell

21 Directory Information Tree (DIT)

Um sich die Funktionsweise eines Verzeichnisdienstes vorstellen zu können, muss zuerst bekannt sein, wie das

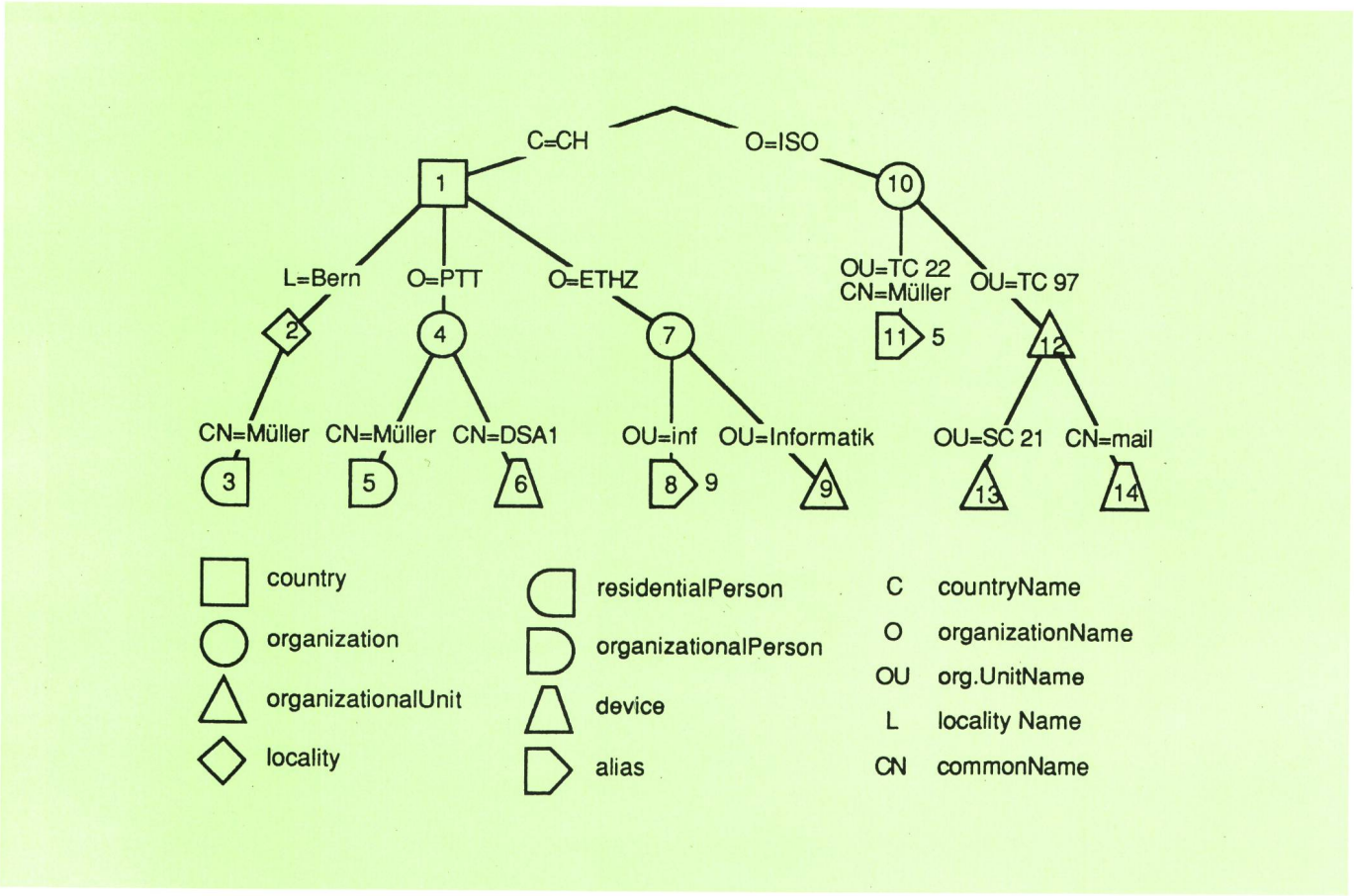

Fig. 1
Beispiel eines Directory Information Tree (DIT)

Verzeichnis strukturiert ist. Wie weiter unten hergeleitet wird, hat es die Form eines Baumes, der *Directory Information Tree* (DIT) genannt wird. *Figur 1* zeigt das Beispiel eines DIT, an dem in den folgenden Abschnitten die verschiedenen Konzepte anschaulich gemacht werden.

Da die Modellierung eines realen Objektes vom Kontext abhängt, in dem es gesehen wird, kann es durch mehrere (logische) Objekte repräsentiert werden. Es ist hier wichtig, zwischen dem realen Bestandteil eines Kommunikationssystems (reales Objekt) und seiner Darstellung durch die ihn charakterisierende Information (logisches Objekt) zu unterscheiden. Die Klammern um das Adjektiv «logisch» deuten an, dass dieses im folgenden fallengelassen wird, wenn das logische Objekt gemeint ist. In Figur 1 beispielsweise wird Herr Müller durch zwei verschiedene Objekte dargestellt. Eintrag 3 enthält die Informationen zum Objekt «Herr Müller als Privatperson, wohnhaft in Bern» und Eintrag 5 jene zum Objekt «Herr Müller als Angestellter der PTT-Betriebe».

22 Objekte

Jedes Objekt hat einen oder mehrere *eindeutige Namen (Distinguished Names)* oder kurz *Namen,* über die es identifiziert wird, und genau eine *Objektinformation.* Der Name dient der Identifikation innerhalb des DIT und die Objektinformation zur Speicherung der relevanten Information.

Sowohl der Name als auch die Objektinformation sind Mengen von *Attributen.* Ein Attribut besteht aus einem *Attributstyp,* der unter anderem den Namen des Attributs, Gleichheitskriterien bei Vergleichen, den Wertebereich usw. festlegt, und einem oder mehreren *Attributswerten.* Figur 1 zeigt die Attribute mit abgekürzten Typenbezeichnungen.

Die Namen der Objekte sind hierarchisch aus Namenskomponenten aufgebaut, die *relativ eindeutige Namen (Relative Distinguished Name, RDN)* heissen und selber aus einem oder mehreren Attributen bestehen. Das Adjektiv «relativ eindeutig» weist darauf hin, dass die Eindeutigkeit nicht global gilt, sondern sich auf einen bestimmten Eintrag bezieht. Das Verzeichnis erhält durch dieses Konstruktionsprinzip eine Baumstruktur, wobei die Knoten Einträge und die Verbindungslinien RDN bezeichnen. Die Empfehlungen schreiben nicht vor, welche Attribute für RDN verwendet werden sollen. Für Figur 1 sind frei gewählte Attribute in den RDN, die sich durch ihre Namen anbieten.

Die Wurzel des DIT ist ein fiktiver Eintrag mit dem leeren Namen. Ein Sohn-Eintrag erhält seinen Namen durch Anfügen eines RDN zum Namen des Vater-Eintrags. Daraus folgt, dass der Name eines Objektes nichts anderes ist als die Sequenz der RDN des Pfades von der Wurzel zum Eintrag des Objektes. In Figur 1 heissen beide RDN, die zu den Einträgen 3 und 5 führen, {CN = Müller} und sind demnach nicht global eindeutig. In bezug auf die Einträge 2 bzw. 4 sind sie jedoch ein-

deutig. Der RDN von Eintrag 10 zu Eintrag 11 besteht aus zwei Attributen: {OU = TC 22, CN = Müller}.

Der Name der Ersterfassung eines Objektes heisst *Hauptname,* die anderen *Synonyme (Alias Names).* Bei der Ersterfassung wird unter dem Hauptnamen ein sogenannter *Objekteintrag* mit der Objektinformation gespeichert. Bei jeder weiteren Erfassung desselben Objektes unter einem Synonym wird ein *Alias-Eintrag* mit einem Verweis auf den Objekteintrag gespeichert. Dieser Verweis ist nichts anderes als der Hauptname des Objektes. In Figur 1 ist der Eintrag 11 ein Alias-Eintrag für den Objekteintrag 5, der zum Objekt «Herr Müller als Angestellter der PTT-Betriebe» gehört. Herr Müller, der in diesem Beispiel bei der ISO eine zusätzliche Führungstätigkeit ausübt, kann nun dank des Alias-Eintrags 11 auch als «Herr Müller, Vorsitzender der Technischen Kommission TC 22 der ISO» angesprochen werden. Herr Müller hat also in diesem Beispiel die drei Namen:

- ({C = CH}, {L = Bern}, {CN = Müller})
- ({C = CH}, {O = PTT}, {CN = Müller}) und
- ({O = ISO}, {OU = TC 22, CN = Müller})

wobei der erste zur Privat- und die beiden letzten zur Geschäftsperson gehören.

Mit Synonymen kann demnach auf ein Objekt mit mehreren Namen zugegriffen werden. Es muss aber betont werden, dass die Information zu diesem Objekt nicht kopiert, sondern nur an einem Ort gespeichert ist.

23 Objektklassen

Um die Objekteinträge einheitlich zu strukturieren, werden Objekte in *Objektklassen (Object Classes)* eingeteilt. Die Objektklasse bestimmt für den Objekteintrag die Attribute, die zum Namen gehören, jene, die zwingend vorhanden sein müssen, und jene, die optional gespeichert werden dürfen. In Figur 1 sind Einträge zu den Objektlassen «country», «organization», «organizational Unit», «locality», «organizationalPerson», «residentialPerson», «device» und «alias» gespeichert.

Eine neue Objektklasse wird aus einer bestehenden abgeleitet, indem sie alle zwingenden und optionalen Attribute der bestehenden Objektklasse erbt und nur noch für zusätzliche Attribute bestimmen muss, ob diese zum Namen gehören, zwingend oder optional sind. Einige vordefinierte Objektklassen und ihre Beziehungen untereinander sind in *Figur 2* dargestellt. Eine Verbindungslinie zwischen zwei Objektklassen bedeutet, dass die untere Objektklasse von der oberen abgeleitet ist.

Dieses Vererbungsprinzip der Attribute erlaubt einen schrittweisen Aufbau komplexerer Objektklassen aus einfacheren. Wie Figur 2 zeigt, gehen die Objektklassen «residentialPerson» und «organizationalPerson» aus «person» hervor. Nun können die Attribute wie z. B. Anrede, die generell für die Beschreibung einer Person benützt werden, in «person» und die spezifischen wie z. B. Hobby oder Bürobezeichnug in «residentialPerson» bzw. «organizationalPerson» definiert werden.

24 DIT-Struktur

Die *DIT-Struktur (DIT Structure)* kann durch ein gerichtetes Netz beschrieben werden *(Fig. 3).* Dessen Knoten sind die Objektklassen, und seine Pfeile verbinden die Objektklassen, die im DIT benachbarte Einträge besitzen dürfen. Zwei Einträge sind benachbart, wenn der eine der Sohn-Eintrag des anderen ist. Die Beziehung benachbarter Einträge kann verschiedene Bedeutung haben. In Figur 1 ist 2 in 1 lokalisiert, 5 bei 4 angestellt, 6 Eigentum von 4, 12 Teil von 10 usw. Die DIT-Struktur bestimmt die erlaubten Ausprägungen des DIT. Die im Anhang B der Empfehlung X.521 vorgestellte DIT-Struktur (Fig. 3 zeigt einen Ausschnitt) ist nur ein Vorschlag und keine Norm, da dieser Anhang formell nicht Bestandteil der Empfehlungen ist.

3 Dienstleistungen aus der Sicht des Benützers

Das Verzeichnissystem ist aus der Sicht des Benützers ein unstrukturiertes System, das seine Dienste in Form von *Operationen* über *Zugriffspunkte* nach aussen anbietet *(Fig. 4).* Ein Benützer – sei dies eine Person oder ein Prozess – nimmt diese Operationen mit Hilfe eines *Benützeragenten (Directory User Agent, DUA)* in Anspruch. Die Kommunikation zwischen DUA und Verzeichnissystem ist verbindungsorientiert. Für den Auf- und den Abbau der Verbindung stehen die Operationen *DirectoryBind* und *DirectoryUnbind* zur Verfügung.

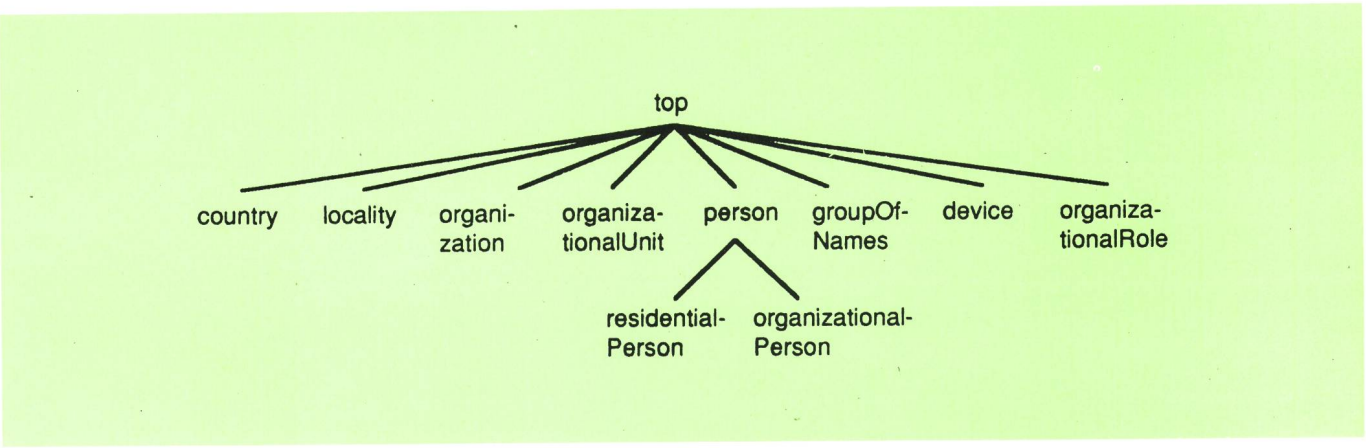

Fig. 2
Objektklassen und ihre Beziehungen

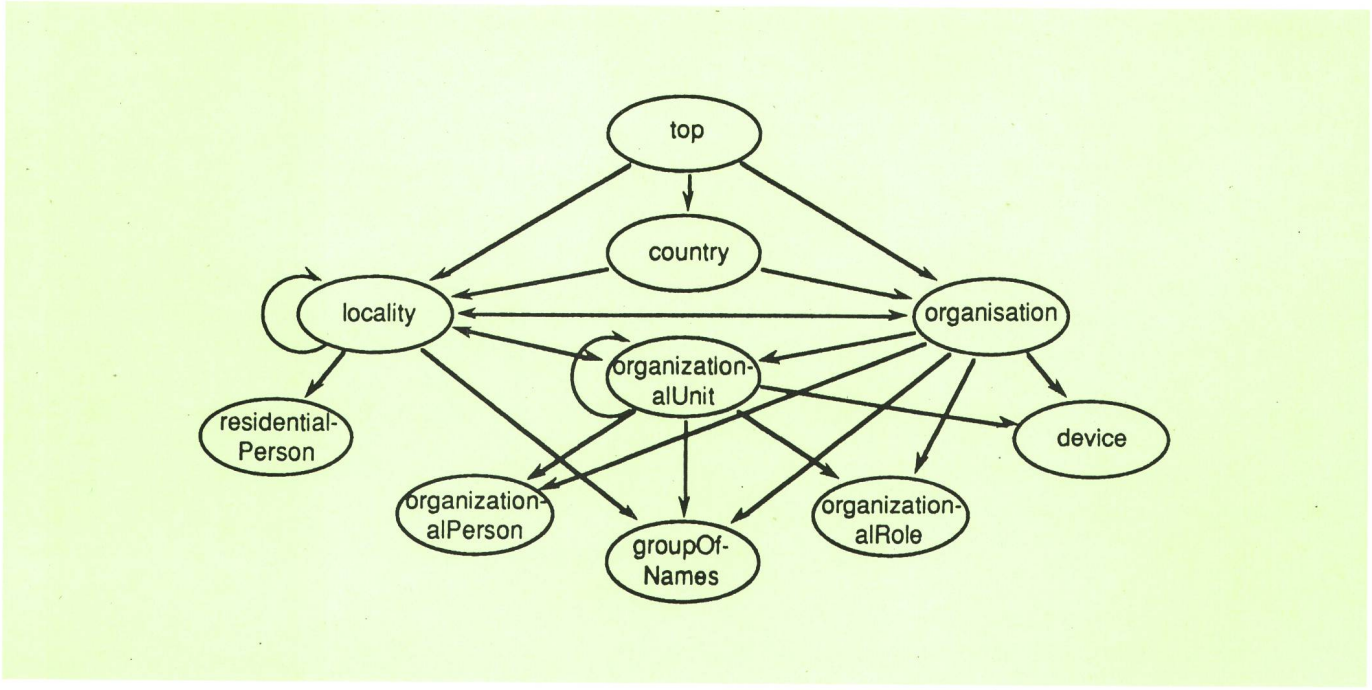

Fig. 3
Eine mögliche DIT-Struktur

31 Operationsklassen

Die angebotenen Operationen können durch ihre Eigenschaften klassifiziert werden. Einerseits unterscheidet man zwischen abfragenden und ändernden Operationen, anderseits können sich Operationen auf einen einzelnen Eintrag oder auf Eintragsgruppen beziehen. Diese beiden unabhängigen Charakterisierungsmerkmale ergeben vier *Operationsklassen (Ports)*. Da X.500 keine ändernden Operationen auf Eintragsgruppen vorsieht, verbleiben noch drei «Ports». Ein Zugriffspunkt zum Verzeichnissystem muss nicht alle drei «Ports» unterstützen. Ein DUA, der mit einem eingeschränkten Zugriffspunkt verbunden ist, kann aber nur einen reduzierten Dienst anbieten. DUA sind demnach konfigurierbar.

Tabelle I. Charakterisierung der «Ports».

	Abfrage	Änderung
Einzeleintrag	readPort	modifyPort
Eintragsgruppe	searchPort	

Tabelle I zeigt die Klasseneinteilung und benennt die drei «Ports». Der *readPort* enthält Abfrageoperationen für einen Eintrag, der *searchPort* solche für Eintragsgruppen und der *modifyPort* Änderungsoperationen für einen Eintrag. Es wird offengelassen, ob in Zukunft für

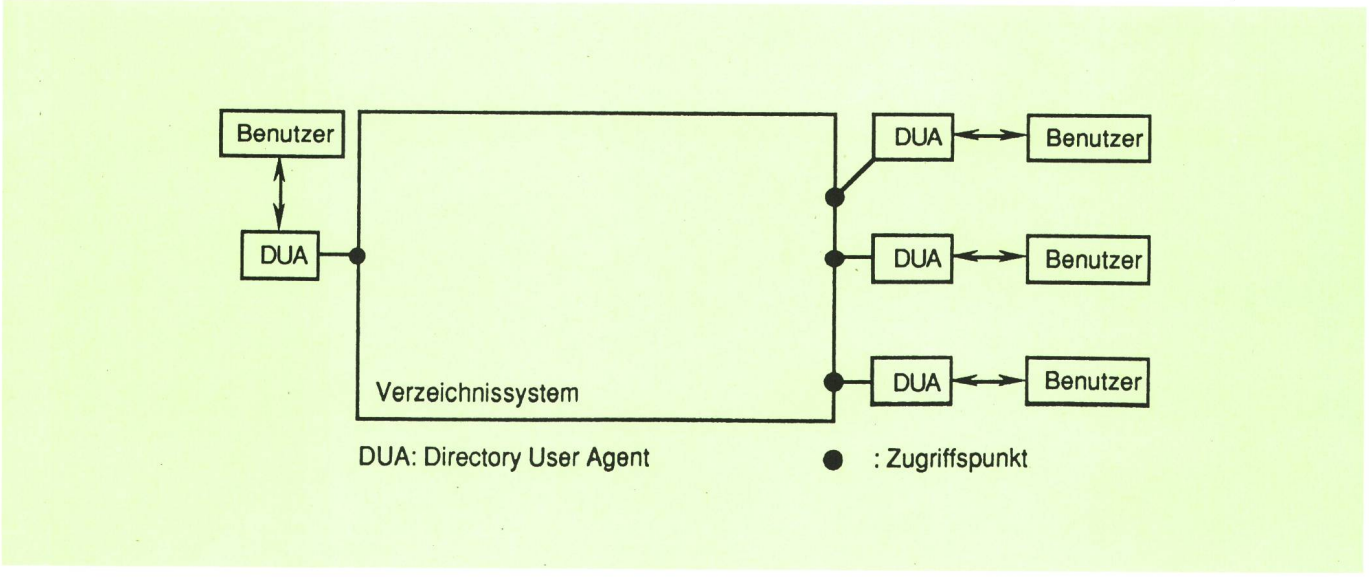

Fig. 4
Zugriff auf das Verzeichnissystem

die Erweiterung der Funktionalität von Verzeichnisdiensten weitere «Ports» hinzukommen. So könnten zusätzliche «Ports» Operationen der vierten Klasse abdecken oder die Verwaltung des Verzeichnisses erlauben.

32 Operationen

Die nachfolgende Liste zeigt die Aufteilung der Operationen auf die drei «Ports». Die Beispiele illustrieren die Verwendung der einzelnen Operationen. Sie ergeben natürlich nur dann einen Sinn, wenn die gewünschte Information auch tatsächlich im Verzeichnis gespeichert ist; andernfalls folgt eine Fehlerantwort.

- Der «readPort» enthält die Operationen *Read, Compare* und *Abandon.* Bei «Read» und «Compare» wird der gewünschte Eintrag über den Namen des gesuchten Objektes identifiziert.
  - «Read» gibt die Objektinformation zurück. Der Benützer kann spezifizieren, welche Attribute relevant sind und gezeigt werden sollen. Mit «Read» kann beispielsweise vom Objekt «Herr Müller als Angestellter der PTT-Betriebe» aus Figur 1 die Adresse für elektronische Post abgefragt werden.
  - «Compare» gibt einen booleschen Wert zurück, der aussagt, ob ein eingegebener Attributswert eines Objekteintrages mit dem entsprechenden Wert im Verzeichnis übereinstimmt. «Compare» ermöglicht unter anderem den Mechanismus zur Benützerauthentifizierung durch das Testen von Passwörtern. Angenommen, Herr Müller hat einen Briefkasten auf dem Rechner eingerichtet, der durch Eintrag 14 repräsentiert wird (Fig. 1). Möchte er die dort neu erhaltenen Briefe lesen, so identifiziert er sich auf diesem Rechner mit Namen und Passwort. Der Rechner kann nun das Passwort mit Hilfe der Operation «Compare» beim Verzeichnisdienst verifizieren, ohne es selber zu kennen.
  - Zum «readPort» gehört auch die Operation «Abandon». Sie beendet Operationen des «readPort» und des «searchPort», falls man an deren Resultat nicht mehr interessiert ist. Es besteht keine Möglichkeit, Änderungen abzubrechen.
- Der «searchPort» besteht aus den Operatioen «List» und «Search». Die gewünschte Eintragsgruppe ist ein Teilbaum des DIT und wird über den Namen seiner Wurzel identifiziert. Der Eintrag dieser Wurzel heisst Basiseintrag.
  - «List» gibt alle Einträge aus, die sich im DIT hierarchisch unmittelbar unter dem Basiseintrag befinden. Mit dieser Operation kann man also eine Liste der Söhne eines Eintrags erzeugen. Unerwünschte Einträge können ausgefiltert werden. Durch wiederholte Ausführung dieser Operation kann der DIT gezielt durchgesehen und geprüft werden. Ein «List» mit dem Basiseintrag 7 (Fig. 1) und der Einschränkung auf Unterorganisationen gibt beispielsweise alle Objekte zurück, die der ETH Zürich unmittelbar untergeordnet sind und zur Objektklasse «organizationalUnit» gehören. Das Resultat dieser Abfrage, basierend auf dem Beispiel-DIT von Figur 1, ist der Eintrag 9 für das Objekt «Departement für Informatik». Eintrag 8 wird ausgefiltert, da er von der Objektklasse «alias» ist.
  - «Search» beschränkt sich nicht nur auf die Söhne, sondern durchsucht den ganzen Unterbaum des Basiseintrages. Auch bei «Search» können unerwünschte Einträge ausgefiltert werden. Diese Operation eignet sich für Yellow-Pages-Abfragen. Eine mögliche Search-Operation auf dem DIT von Figur 1 ist die Suche nach allen im Verzeichnis gespeicherten Rechnern (Objektklasse «device») in der Schweiz. Der Basiseintrag dieser Operation ist Eintrag 1 und dessen Resultat der Eintrag 6 für den von den Schweizerischen PTT betriebenen Rechner DSA1.
- Der «modifyPort» enthält die Operationen *AddEntry, RemoveEntry, ModifyEntry* und *ModifyRDN.* Der gewünschte Eintrag wird über seinen Namen identifiziert.
  - «AddEntry» und «RemoveEntry» dienen dem Hinzufügen und dem Löschen ganzer Einträge. Sie können nur auf Blatteinträge des DIT angewendet werden; das heisst, das eingefügte oder das zu löschende Objekt darf nicht der Vater-Eintrag irgendeines anderen Eintrags sein. Soll beispielsweise in Figur 1 die Abteilung, in der Herr Müller (Eintrag 5) arbeitet, als Unterorganisation der PTT (Eintrag 4) in den DIT aufgenommen werden, so kann sie nicht direkt dazwischen eingefügt, sondern muss als Blatt an den Eintrag 4 angehängt und danach Eintrag 5 umgehängt werden. Ist Eintrag 5 kein Blatteintrag, so wird dieses Vorhaben noch komplizierter.
  - «ModifyEntry» erlaubt das Hinzufügen und das Löschen ganzer Attribute, das Hinzufügen, das Löschen und das Ersetzen von Attributswerten sowie das Ändern eines Alias-Eintrags.
  - «ModifyRDN» ermöglicht die Änderung der letzten Namenskomponente des Namens eines Eintrages. Auch hier muss der Eintrag ein Blatteintrag des DIT sein. Diese Operation entspricht einem Löschen und einem nachfolgenden Anfügen unter anderem Namen an denselben Vater-Eintrag. Die ModifyRDN-Operation ist nach Ermessen des Verfassers unmotiviert, da sie einerseits die Funktionalität nicht erweitert und anderseits die Lebensdauer von Namen unerwünschterweise verkürzt.

33 Operationsparameter und ihre Informationstypen

Jede Operation besitzt einen Namen und die drei Parameter *Argument, Resultat* und *Fehler.* Argument ist der Eingabeparameter, der die Operation genauer spezifiziert und sie damit meist auch einschränkt. Resultat und Fehler sind die Ausgabeparameter, die sich gegenseitig ausschliessen, da sie ein erfolgreiches bzw. ein erfolgloses Beenden einer Operation anzeigen. Die Struktur der drei Parameter ist durch sogenannte *Informationstypen* beschrieben. Die wichtigsten Informationstypen sind *Eintragsinformation, Steuerelement, Filter* und *Sicherheitsparameter:*

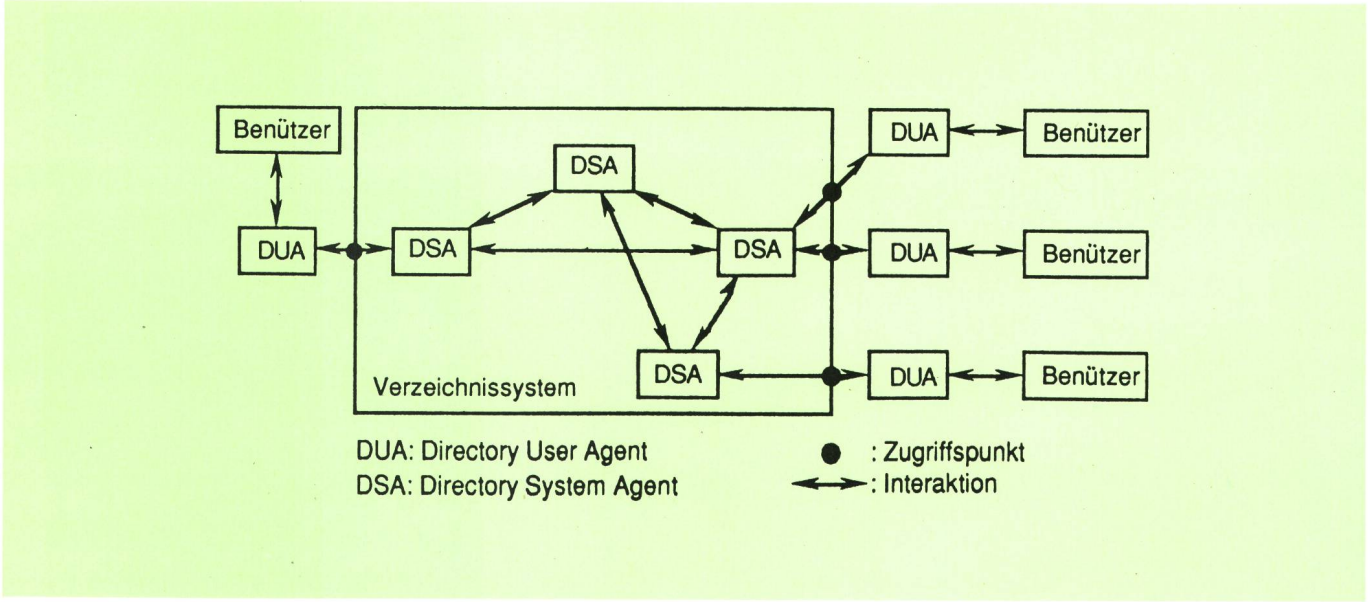

Fig. 5
Komponenten des Verzeichnissystems

- Die Eintragsinformation enthält die Einträge, wie sie im Resultatsparameter dem Benützer zurückgegeben werden.
- Mit den Steuerelementen werden Abfragen in bezug auf die folgenden Kriterien spezifiziert: Zeitlimit, Umfangslimit, Priorität der Abfrage, Einschränkung auf das lokale Verzeichnis usw. Die Steuerelemente spezifizieren auch, ob Synonyme dereferenziert (aufgelöst) werden dürfen. Meist ist der Benützer an der Objektinformation interessiert, weshalb die Dereferenzierung normalerweise automatisch erfolgt. In einzelnen Fällen ist jedoch der Alias-Eintrag selbst das Ziel einer Operation. Soll beispielsweise in der Figur 1 der Alias-Eintrag 11 gelöscht werden, so muss die Dereferenzierung unterbunden werden, da sonst der Eintrag 5 gelöscht würde.
- Filter erlauben bei Search-Operationen die Resultatemenge a priori einzuschränken, indem zusätzliche Bedingungen an die gesuchten Einträge geknüpft werden. Beim obigen Beispiel zur Search-Operation kann mit einem Filter zusätzlich präzisiert werden, dass nur jene Rechner interessieren, die für die Meldungsübermittlung verwendet werden. Voraussetzung ist natürlich, dass diese Information beispielsweise im Attribut «description» festgehalten ist.
- Auf die Sicherheitsparameter kann im Rahmen dieses Artikels nicht eingegangen werden.

4 Aspekte der verteilten Architektur

41 Funktionelle Verteilung

Wie in der Einführung zu diesem Artikel erwähnt, sind Verzeichnissysteme verteilt, da sie ein umfangreiches Datenvolumen umfassen, das von den verschiedensten Institutionen unterhalten wird. Weitere Gründe für die Verteilung sind gewünschte hohe Verfügbarkeit und ortsunabhängige Funktion. Die Komponenten des Verzeichnissystems heissen *Directory System Agents (DSA).* Sie erlauben einen unabhängigen Zugang von verschiedenen Orten und erbringen gemeinsam den gewünschten Dienst *(Fig. 5).* Wie im folgenden erläutert wird, speichert und verwaltet jeder DSA nur einen Teil des Verzeichnisses. Um Benützeraufträge zu beantworten, die sich auf die in anderen DSA gespeicherten Informationen beziehen, müssen die DSA untereinander verbunden sein. Nur so kann ein DSA einen Benützerauftrag in Teilaufträge zerlegen und andere DSA mit deren Bearbeitung beauftragen.

Ein DSA erbringt seinen Dienst gegenüber DUA und DSA. Er offeriert also die Operationen, die er den DUA anbietet, in analoger Weise auch anderen DSA. Die zu «Bind» und «Unbind» analogen Operationen für den Auf- und den Abbau von Verbindungen zwischen zwei DSA heissen «DSABind» und «DSAUnbind». Die Operationen für die DSA untereinander sind zu «distributedPorts» zusammengefasst. Ein «searchPort» beispielsweise bietet den DUA die Operationen «List» und «Search» an, während ein «distributedSearchPort» anderen DSA die Operationen «DistributedList» und «Distributed Search» anbietet. Die Operationen der «distributedPorts» benötigen als Eingabeparameter einige zusätzliche Informationstypen, um die bei der Verteilung entstehenden Probleme zu behandeln. Die analogen Operationen unterscheiden sich also nur syntaktisch, nicht aber semantisch.

Wichtige zusätzliche Informationstypen sind *Originator, Operation Progress* und *Trace Information:*

- «Originator» gibt über den Auftraggeber einer Operation genauere Auskunft, damit die ermittelten Teilresultate korrekt an diesen zurückgesendet werden können.
- Anhand von «Operation Progress» können DSA, die mit Teilaufgaben von Operationen betraut werden, sich über die schon erledigten Teilaufgaben informieren.

– «Trace Information» hält den Weg eines Benützerauftrags innerhalb des Verzeichnissystems fest und erkennt so Verarbeitungszyklen oder Inkonsistenzen im DIT.

42 Datenverteilung

Während sich der Verzeichnisdienstbenützer nicht für die Datenverteilung interessiert, stellt sich vom Standpunkt des Verzeichnissystems die Frage, wie man das Verzeichnis partitioniert, die dabei entstehenden Partitionen den DSA zuteilt und das Wissen über die Verteilung verwaltet.

Partitionierung des DIT in Kontexte

Ein *Kontext (Namenskontext)* ist ein Unterbaum des DIT, der sich nicht zwingend bis zu den Blattknoten erstreckt. Den Namen des Wurzeleintrages dieses Unterbaumes bezeichnet man mit *Präfix (Kontextpräfix).* Die Kontexte dienen einer hierarchischen Partitionierung des DIT, das heisst sie überdecken ihn vollständig, überlappen sich jedoch nicht *(Fig. 6).* Ein Kontext wird von genau einem DSA verwaltet, jedoch kann ein DSA mehrere Kontexte verwalten. Der Kontext mit dem leeren Präfix heisst *Wurzelkontext,* seine hierarchisch unmittelbar untergeordneten Kontexte sind *First-Level-Kontexte (FL-Kontexte)* und die DSA, die solche Kontexte speichern, *First-Level-DSA (FL-DSA).*

Figur 6 zeigt vier Kontexte (a, b, c und d), die auf drei DSA (A, B und C) verteilt sind. Die Präfixe lauten ({C = CH}) für a, ({C = CH}, {O = PTT}) für b, ({C = CH}, {O = ETHZ}) für c und ({O = ISO}) für Kontext d; a und d sind FL-Kontexte und A und C FL-DSA.

Obwohl die Empfehlungen keine Datenreplizierung unterstützen, können aufgrund bilateraler Absprachen Teile des DIT redundant gespeichert werden. X.500 trifft dazu zwei Vorkehrungen: Erstens können Aufträge spezifizieren, ob sich ihre Beantwortung auf Kopien stützen darf, und zweitens geben Antworten an, ob sie aufgrund von Originalen oder von Kopien entstanden sind.

Referenzen für die Kontextverteilung

Das Wissen über die Verteilung der Kontexte wird mittels *Referenzen* ausgedrückt. Eine Referenz gehört zu einem Kontext und bringt diesen zu einem anderen Kontext in eine hierarchische Beziehung. Es werden fünf Referenztypen unterschieden:

– *Unterreferenzen (Subordinate References)* werden für unmittelbar untergeordnete Kontexte verwendet, deren RDN bekannt ist. Sie bestehen aus diesem RDN sowie aus dem Namen und der Präsentationsadresse des DSA, der diesen Kontext speichert. In Figur 6 sind die Kontexte b und c dem Kontext a untergeordnet. Da der Kontext a die RDN kennt, die zu b und c führen (Fig. 1: {O = PTT} und {O = ETHZ}), besitzt er die beiden Unterreferenzen [1, {O = PTT}, DSA B, <Adresse B>] und [1, {O = ETHZ}, DSA B, <Adresse B>]. Die Zahl 1 besagt, dass die Referenz bei Eintrag 1 ansetzt.

– *Nichtspezifische Unterreferenzen (Non-Specific Subordinate References)* bestehen aus dem Namen und der Präsentationsadresse des DSA, der einen unmittelbar untergeordneten Kontext speichert, dessen RDN nicht bekannt ist. Würde Kontext a in Figur 6 den RDN zu Eintrag 7 nicht kennen, so speicherte er eine nichtspezifische Unterreferenz zu Kontext c von der Form [1, DSA B, <Adresse B>].

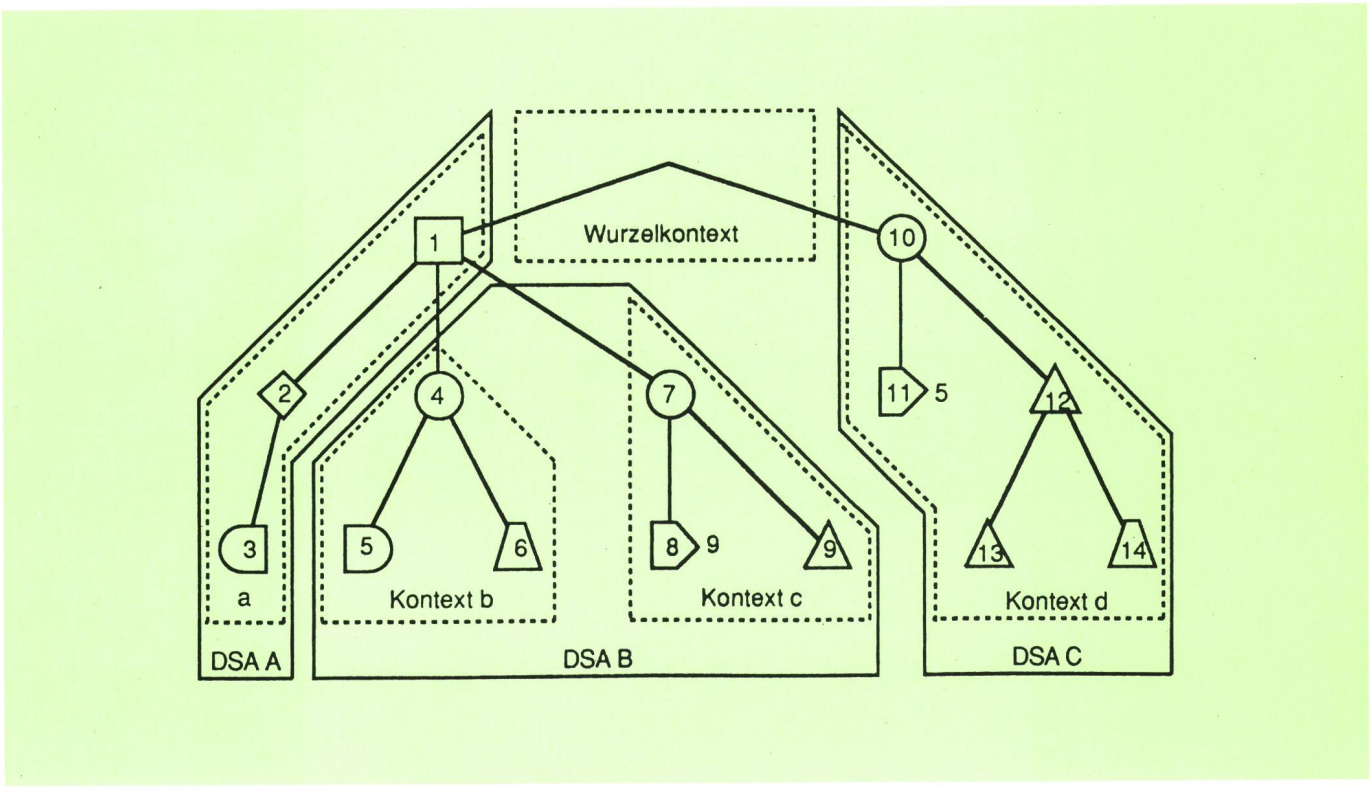

Fig. 6
Hypothetische Zerlegung des DIT nach Figur 1

- *Überreferenzen (Superior References)* enthalten den Namen und die Präsentationsadresse des DSA, der den unmittelbar übergeordneten Kontext speichert. FL-Kontexte und der Wurzelkontext besitzen keine Überreferenz. Die Kontexte b und c in Figur 6 besitzen die Überreferenz [DSA A, <Adresse A>].
- *Querreferenzen (Cross References)* dienen Optimierungszwecken. Sie setzen sich aus dem Präfix eines Kontextes, der nicht unmittelbar über- oder untergeordnet ist, sowie aus dem Namen und der Präsentationsadresse des DSA zusammen, auf dem der entsprechende Kontext gespeichert ist. Kontext d in Figur 6 kann eine Querreferenz zu Kontext b speichern, um direkt auf diese Information zugreifen zu können, ohne über den Kontext a gehen zu müssen. Die Referenz sieht so aus: [({C = CH}, {O = PTT}), DSA B, <Adresse B>].
- *Interne Referenzen (Internal References)* dienen dem Auffinden der Einträge innerhalb eines Kontextes und bestehen aus dem RDN dieser Einträge sowie aus einem Verweis auf die lokale Datenbasis. Letzterer liegt ausserhalb der Empfehlungen. Für jeden Eintrag existiert genau eine interne Referenz. Damit der RDN zur Identifikation eines Eintrages genügt, muss der lokale Aufbau des DIT innerhalb des Kontextes bekannt sein. Die Speicherung dieser Information liegt auch ausserhalb von X.500. Die interne Referenz zu Eintrag 3 in Figur 6 lautet [2, {CN = Müller}, <Adresse 3>]. Die Zahl 2 gibt an, dass Eintrag 3 Sohn von Eintrag 2 ist. Ihre Darstellung ist nicht durch die Empfehlungen vorgeschrieben.

Das Verzeichnissystem muss so organisiert sein, dass jeder DUA (über seinen DSA) auf jeden beliebigen Eintrag zugreifen kann. Deshalb muss jeder DSA anhand der Referenzen seiner Kontexte entscheiden können, auf welchem DSA ein Eintrag gespeichert ist. Eine Referenz zu einem gesuchten Kontext muss nicht zwingend explizit vorhanden sein, sondern kann auch durch einen *Referenzpfad (Reference Path)* konstruiert werden. Um garantieren zu können, dass jederzeit ein Pfad gefunden wird, muss ein DSA mindestens eine Überreferenz und alle Unterreferenzen (die nichtspezifischen eingeschlossen) kennen. Ein DSA kann zusätzlich beliebig viele Querreferenzen speichern, um seine Leistungsfähigkeit zu erhöhen.

Auf diese Weise ist immer ein Pfad vorhanden, der eventuell über den Wurzelkontext führt. Dieser spielt dabei eine Sonderrolle, da er nicht auf einem einzigen DSA gespeichert ist, sondern vielmehr auf jedem FL-DSA repliziert wird, so dass jeder FL-Kontext durch die Unterreferenzen des Wurzelkontextes alle anderen FL-Kontexte kennt. Das Wissen über FL-Kontexte wird demnach redundant gespeichert. Dieser Ansatz beruht auf der Annahme, dass ein weltweites System eine kleine Anzahl von FL-DSA aufweisen wird, beispielsweise einen pro Land, die zusammen einen leistungsfähigen Datendurchsatz, eine hohe Zuverlässigkeit und eine flexible, verteilte Autorität in der Verwaltung des ganzen Verzeichnisses aufweisen.

43 Operationelle Verteilung

Ein Benützerauftrag kann oft nicht lokal behandelt werden, da die Information dazu auf mehreren DSA verteilt ist. Dieser ist demnach keine atomare Einheit, sondern enthält mehrere Phasen der Bearbeitung. Dabei müssen gemäss den lokal vorhandenen Referenzen weitere DSA einbezogen werden, um Teilaufträge weiterzuleiten. Aus den verschiedenen Referenztypen lassen sich mehrere Interaktionsmodi zwischen DSA ableiten.

Interaktionsmodi zwischen DSA

Man unterscheidet die drei Interaktionsmodi *Chaining, Multicasting und Referral (Fig. 7).* Die Zahlen in der Figur geben den zeitlichen Ablauf an.

- Ein DSA benützt «Chaining», wenn er anhand seiner Unter-, Über- oder Querreferenzen schlüssig entscheiden kann, welcher DSA mit der Bearbeitung von Teilaufträgen in Anspruch zu nehmen ist.
- «Multicasting» kommt dann zur Anwendung, wenn ein DSA aufgrund von nichtspezifischen Unterreferenzen nicht schlüssig festlegen kann, welcher DSA über die

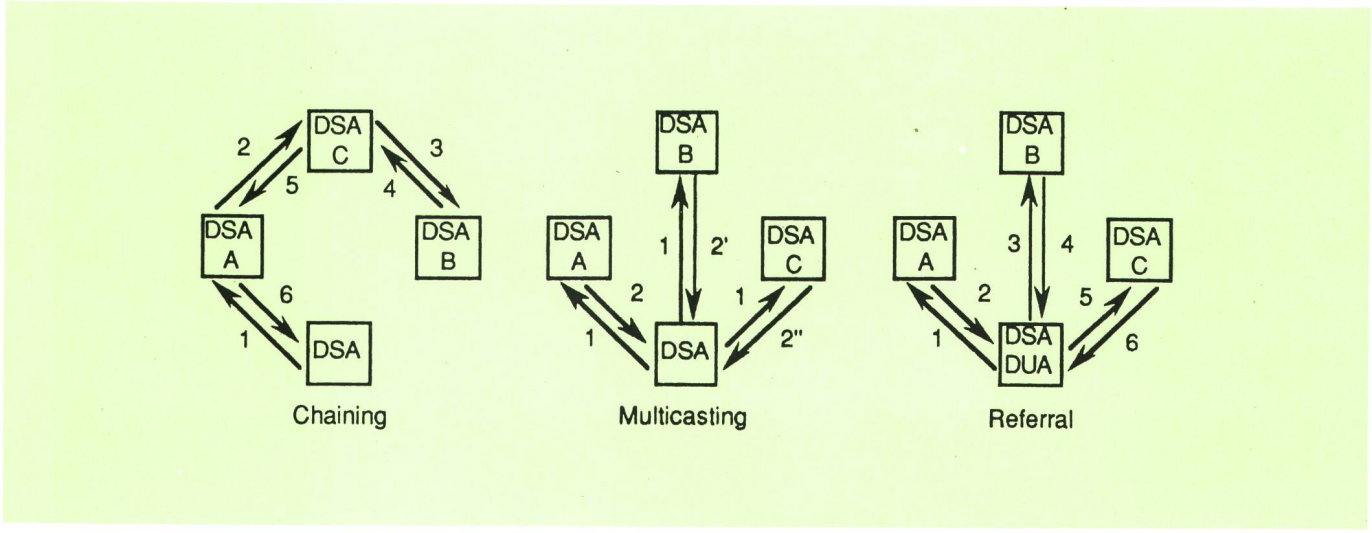

Fig. 7
Interaktionsmodi zwischen DSA

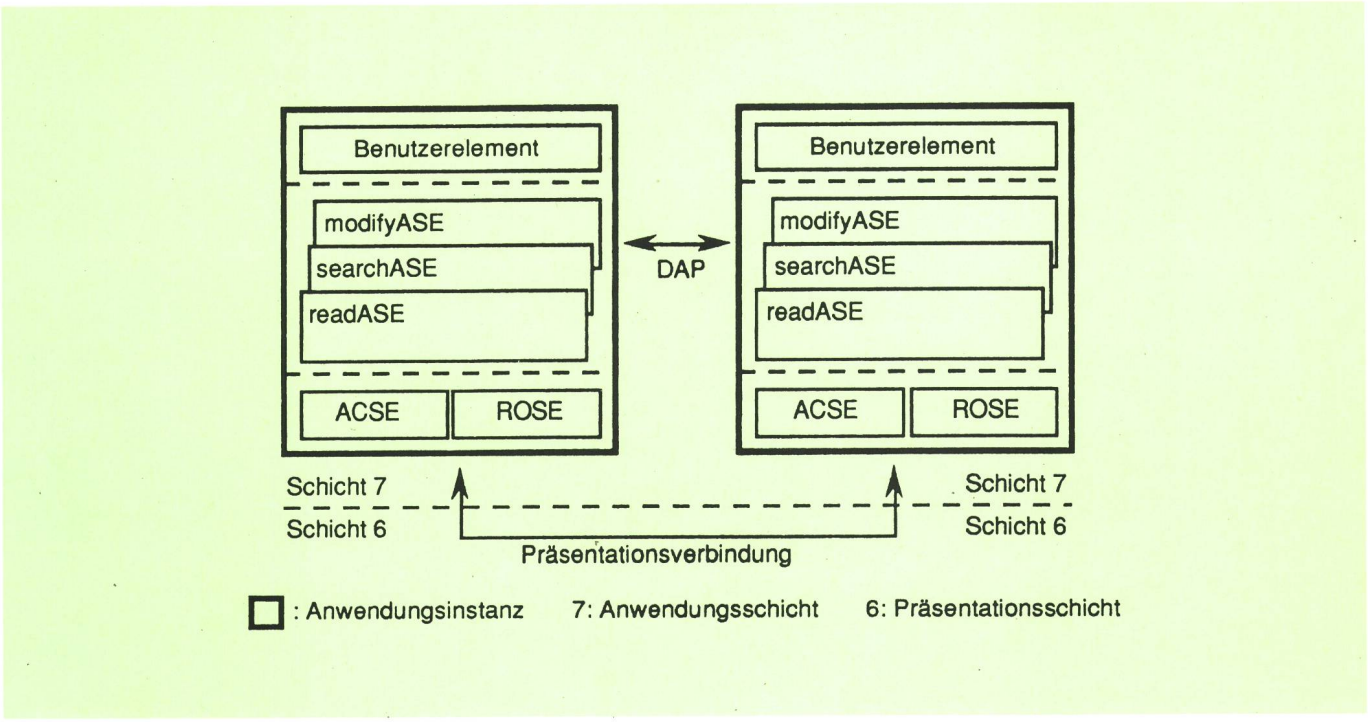

Fig. 8
Einbettung in das OSI-Referenzmodell: DAP

notwendigen Daten für die Bearbeitung von Teilaufträgen verfügt oder wenn er im voraus weiss, dass mehrere DSA involviert sind.

- Ein «Referral» ist immer eine Antwort auf einen Auftrag im Chaining-Modus oder im Multicasting-Modus, der nicht oder nur unvollständig verarbeitet werden konnte. Die Antwort enthält dann eine Referenz zu einem weiteren DSA, der wieder im Chaining-Modus oder im Multicasting-Modus beauftragt werden muss. Im Referral-Modus kann interessanterweise auch ein DUA die Koordination übernehmen.

Die Wahl des Interaktionsmodus ist im allgemeinen den DSA überlassen. Der Benützer kann allerdings «Chaining» verbieten, so dass ein beauftragter DSA mit einem «Referral» oder einer Fehlermeldung antworten muss.

Phasen der Bearbeitung eines Benützerauftrags

Die Bearbeitung eines Benützerauftrags lässt sich chronologisch in die drei Phasen *Namensauflösung, Evaluation* und *Resultatekollektierung* unterteilen.

- Die *Namensauflösung (Name Resolution)* ermittelt für den Namen eines Eintrages anhand der internen Referenzen, ob der Eintrag lokal gespeichert ist. Ist dies der Fall, so ist die Namensauflösung beendet, und der Auftrag kann in die nächste Phase eintreten, andernfalls wird mit den anderen Referenzen der DSA gesucht, der den Eintrag verwaltet.
- Die *Evaluation* wird dann aufgerufen, wenn die Namensauflösung eine interne Referenz gefunden hat. Sie führt die eigentliche Operation aus. Da beispielsweise bei den Operationen «List» und «Search» mehrere Einträge betroffen sein können, umfasst die Evaluationsphase für eine Operation unter Umständen mehrere DSA. Jeder involvierte DSA liefert dabei Teilresultate für die weitere Verarbeitung.
- Sobald Teilresultate der Evaluation verfügbar sind, werden diese in der Phase der *Resultatekollektierung (Result Merging)* zu einem Gesamtresultat zusammengefügt und dieses dem Auftraggeber zurückgesendet.

5 Einbettung in das OSI-Modell

In Abschnitt 3 wurden die Dienste vorgestellt, die das Verzeichnissystem einem Benützer über einen DUA anbietet. Sind der DUA des Benützers und der das Verzeichnissystem repräsentierende DSA in verschiedenen realen offenen Systemen lokalisiert, so ist deren Kommunikation durch das *Verzeichniszugangsprotokoll (Directory Access Protocol, DAP)* definiert. In Abschnitt 4 wurden jene Aspekte erläutert, die für die Weiterleitung von Teilaufträgen unter DSA wichtig sind. Wenn zwei kommunizierende DSA in verschiedenen realen offenen Systemen lokalisiert sind, dann ist deren Kommunikation durch das *Verzeichnissystemprotokoll (Directory System Protocol, DSP)* definiert.

Im Sinne von OSI sind DUA und DSA Anwendungsprozesse und ihre OSI-relevanten Teile Anwendungsinstanzen. *Figur 8* zeigt die Kommunikation zwischen einem DUA und einem DSA, die durch das DAP definiert ist. *Figur 9* zeigt die analoge Situation der Kommunikation zweier DSA, die durch das DSP definiert ist.

Die Anwendungsinstanzen von DUA und DSA können grob in drei Teile aufgeteilt werden. Diese heissen: *Benützerelement, verzeichnisspezifische Anwendungsdienstelemente* und *allgemeine Anwendungsdienstelemente.* Ein Anwendungsdienstelement (Application Service Element, ASE) besteht aus einer Menge von Funktionen, die den verbundenen Anwendungsinstanzen eine bestimmte OSI-konforme Zusammenarbeit ermöglichen.

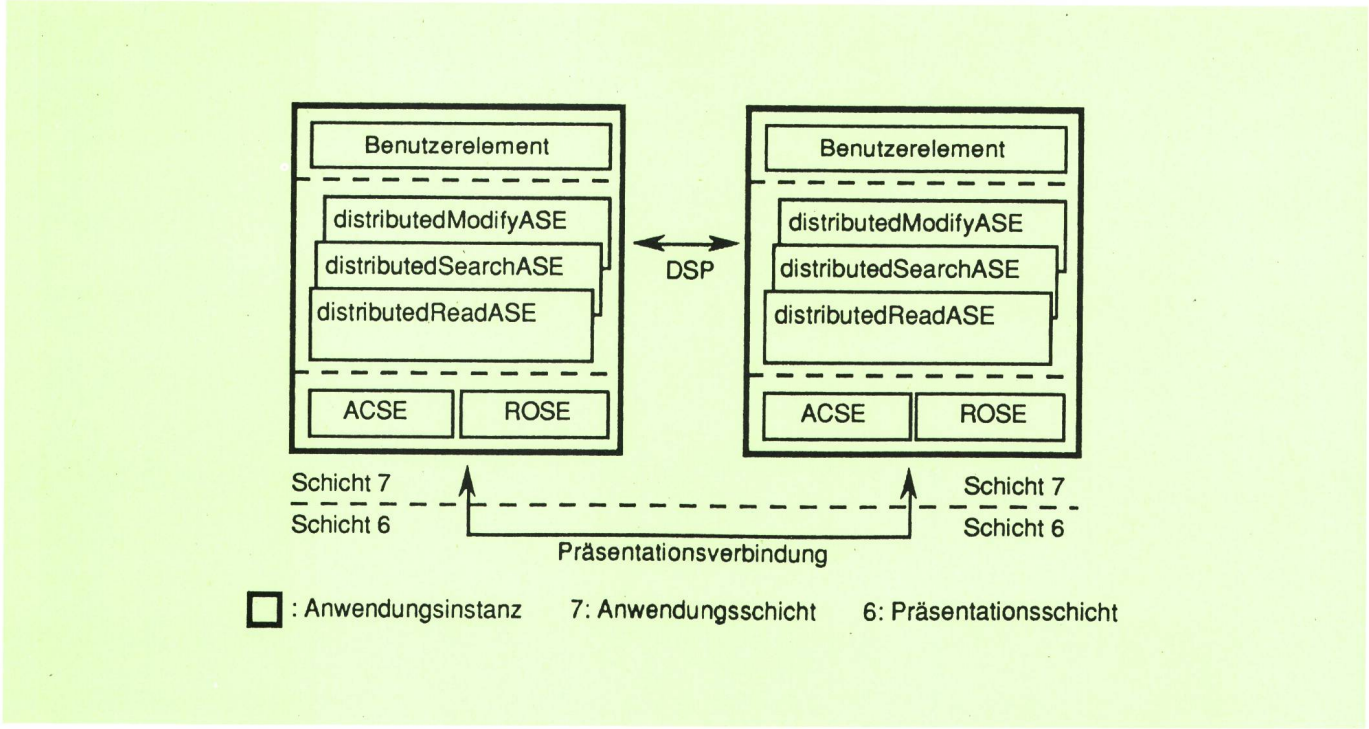

Fig. 9
Einbettung in das OSI-Referenzmodell: DSP

Das Benützerelement ist die Schnittstelle zwischen der OSI-Umgebung und der Umgebung des realen Systems. Hier stehen einem Benützer die Verzeichnisdienstoperationen zur Verfügung.

Die verzeichnisspezifischen ASE widerspiegeln die Portaufteilung und sind deshalb entsprechend benannt. Beim DAP heissen sie *readASE, searchASE* und *modifyASE,* beim DSP *distributedReadASE, distributedSearchASE* und *distributedModifyASE.* Da die verzeichnisspezifischen ASE ihre ganze Funktionalität auf die Dienste der allgemeinen ASE abbilden, bilden das DAP und das DSP lediglich einen Satz von syntaktischen und semantischen Definitionen dieser Abbildung.

Allgemeine ASE stellen für die verschiedensten Anwendungen unterstützende Dienste bereit. Die Agenten des Verzeichnisdienstes benützen zwei davon: *Association Control Service Element (ACSE)* und *Remote Operations Service Element (ROSE).*

51 Beziehung zu X.400

Message Handling Systems (MHS) werden voraussichtlich die ersten und wichtigsten Benützer von Verzeichnisdiensten sein. Erst X.400 (88) schafft jedoch die Voraussetzung dafür, dass der Verzeichnisdienst direkt oder indirekt genutzt werden kann, da sie den Begriff *O/R-Name* umdefiniert. Ein O/R-Name bezeichnet in X.400 (84) eine Adresse, für die neu der Ausdruck *O/R-Adresse* verwendet wird. In X.400 (88) kann nun ein O/R-Name entweder eine O/R-Adresse, einen Verzeichnisnamen oder beides enthalten. Verzeichnisnamen sind im Sinne von X.500 die Namen von Objekten. Aus der Neudefinition von O/R-Namen gehen mehrere Möglichkeiten hervor, wie ein MHS den Verzeichnisdienst benützen kann.

- *Benützerfreundliche Namen:* Absender und Empfänger von Meldungen können mit einem benützerfreundlichen Namen bezeichnet werden. Besteht ein O/R-Name nur aus dem Namen, so muss dieser mit Hilfe des Verzeichnisdienstes mit der entsprechenden Adresse ergänzt werden. Die Aufteilung des O/R-Namens in eine Namens- und eine Adresskomponente ist sinnvoll, da die erstere vom Benützer und die letztere vom Message Transfer System (MTS) verwendet wird. Gemäss den X.400-Empfehlungen können nämlich Namen in der Kommunikation zwischen einem User Agent (UA) und einem Message Store (MS), einem MS und einem Message Transfer Agent (MTA) und zwischen einem UA und einem MTA verwendet werden, nicht aber zwischen zwei MTA. Das bedeutet, dass der UA des Absenders, sein MS oder der erste MTA des MTS die entsprechende Adresse im Verzeichnis suchen und einsetzen muss.

- *Verteilerlisten (Distribution Lists, DL):* Eine Gruppe von Empfängern kann in einer DL zusammengefasst werden. Sie hat eine O/R-Adresse und zusätzlich einen Namen, falls sie vom Verzeichnisdienst verwaltet wird. Steht beim Recipient-Feld einer Meldung der Name einer Verteilerliste, so muss dieser spätestens beim ersten MTA mit der entsprechenden O/R-Adresse ergänzt und beim verantwortlichen MTA *(Expansion Point)* durch die O/R-Adressen der DL-Mitglieder ersetzt werden. Beide Aktionen folgen mit Hilfe des Verzeichnisdienstes. Die erhaltenen O/R-Adressen können wiederum Verteilerlisten bezeichnen. So kann eine bestimmte O/R-Adresse mehrmals oder die O/R-Adresse der ursprünglichen Verteilerliste wieder enthalten sein. Aus diesem Grund werden alle *Notifications* und *Interpersonal Message Receipts* an den verantwortlichen MTA zurückgesandt und nicht direkt an den eigentlichen «Originator» der Mel-

dung. Es ist dann die Aufgabe dieses MTA, sie an den «Originator» oder an den Administrator der Verteilerliste weiterzuleiten.

- *Mächtigkeit der MHS-Komponenten:* Die Dienste, die ein UA, ein MS oder ein MTA unterstützt, können in dessen Verzeichniseintrag gespeichert werden. Erfragt man die Mächtigkeit einer Komponente direkt beim Verzeichnisdienst, so muss keine *Probemeldung (Probe Message)* an diese gesendet werden, womit das MTS entlastet wird.
- *Gegenseitige Authentifizierung:* Vor der eigentlichen Kommunikation authentifizieren sich die MHS-Komponenten gegenseitig. In Abschnitt 6 wird dieser Ablauf genauer erläutert.
- *Interaktive Nutzung:* Der X.400-Benützer kann den Verzeichnisdienst direkt befragen, um Empfänger und ihre O/R-Adressen zu finden. Hat er den Namen, so kann er mit der Read-Operation die entsprechende O/R-Adresse erhalten. Besitzt er nur Teile des Namens oder andere Information, so muss er die Search-Operation anwenden.

Figur 10 zeigt das funktionale Modell der Beziehung zwischen X.500 und X.400. Es ist erkennbar, dass die Verwendung eines Verzeichnissystems eine lokale Angelegenheit einzelner MHS-Komponenten ist und deshalb weder einen Einfluss auf die MHS-Protokolle hat noch die Anbieter oder die Verwalter anderer Komponenten zwingt, ebenfalls diese Dienste zu integrieren. Das Modell fordert kein globales Verzeichnissystem. Jede Komponente, die von einem Verzeichnissystem profitieren will, muss einen lokalen DUA für den Zugriff auf dieses System besitzen, da es kein Protokoll zwischen X.500- und X.400-Komponenten gibt. In der Figur 10 sind die X.400-Komponenten, die vom Verzeichnisdienst Gebrauch machen, hervorgehoben.

Aus dem funktionalen Modell lassen sich verschiedene Komponentenkonfigurationen ableiten. Ist ein DUA auf einem UA-System lokalisiert, so stellt dies eine Art intelligentes Terminal dar, das zusätzlich zur Meldungsübermittlung interaktive Abfragen zum Verzeichnissystem erlaubt. Ist ein DUA in einem MS-System integriert, so wird damit die Anzahl DUA reduziert, da ein MS typischerweise mehrere UA bedient. Inkorrekte Meldungen können noch abgefangen werden, bevor sie an das MTS übergeben werden. Als letzte Variante kann der DUA in einem MTA-System plaziert werden. Dies hat den Nachteil, dass die zu bearbeitenden Meldungen im Moment, in dem das Verzeichnis abgefragt wird, bereits übermittelt sind. Vorteilhaft wäre in diesem Fall jedoch die Tatsache, dass der DUA gleichzeitig für die Evaluation von Namen und von Verteilerlisten genutzt werden kann. Eine tatsächliche Ausführung wird wahrscheinlich mehrere der in Figur 10 aufgezeigten Komponenten auf einem System vereinigen, so dass beispielsweise ein DSA und ein MTA mit einem DUA gekoppelt und auf demselben System lokalisiert werden.

6 Sicherheit

Für Verzeichnisdienste können in bezug auf die Sicherheit zwei Aufgabenbereiche unterschieden werden: Einerseits muss die Kommunikation eines Benützers mit dem Verzeichnissystem sicher gestaltet werden, und anderseits kann der Verzeichnisdienst Mechanismen bereitstellen, die den Informationsaustausch unter beliebi-

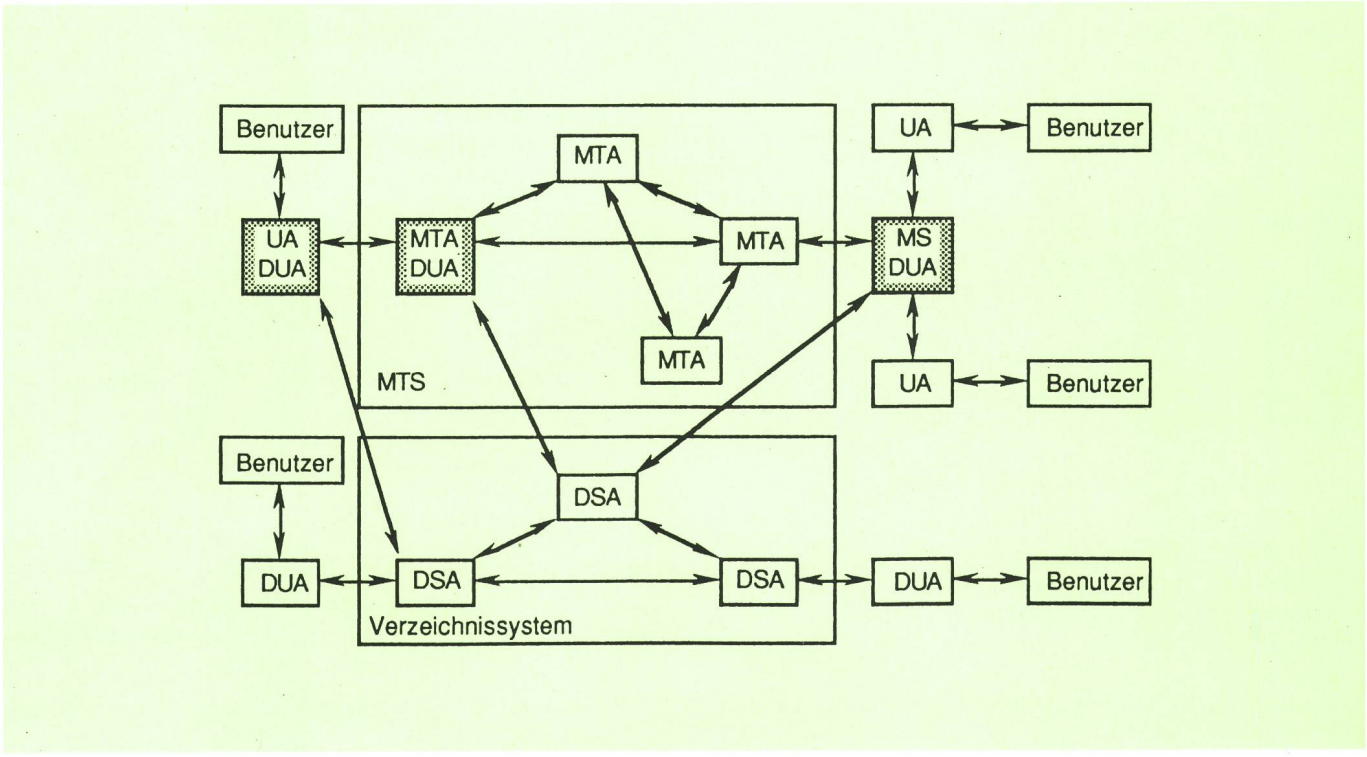

Fig. 10
Funktionales Modell der Beziehung zwischen X.500 und X.400

gen Teilnehmern eines Kommunikationssystems sichern helfen:

- Bei der *Kommunikation mit dem Verzeichnissystem* müssen sich Benützer und Verzeichnissystem gegenseitig authentifizieren, damit einerseits der Benützer die Gewissheit erhält, auch tatsächlich mit dem gewünschten Dienstanbieter in Verbindung zu stehen, und anderseits das Verzeichnissystem die Zugriffsrechte auf das Verzeichnis kontrollieren und die beanspruchten Leistungen verrechnen kann. Daneben müssen das Abhorchen, das Duplizieren, das Fehlleiten und das Abändern der Information bei deren Austausch verhindert werden.
- Beim *Informationsaustausch unter beliebigen Teilnehmern* eines Kommunikationssystems bestehen dieselben Sicherheitsansprüche, die mit den Begriffen *Teilnehmerauthentifizierung* und *Informationsauthentifizierung* zusammengefasst werden können. Das Verzeichnissystem kann dazu Passwörter, Zertifikate usw. speichern und die Operationen mit Sicherheitsparametern versehen.

Der zweite Aufgabenbereich ist allgemeiner, da das Verzeichnissystem selber als ein Teilnehmer des Kommunikationssystems aufgefasst werden kann. Das in X.500 definierte Sicherheitskonzept unterstützt diesen allgemeinen Fall. Es unterscheidet dabei zwischen zwei verschiedenen Sicherheitsstufen. Die *Einfache Authentifizierung (Simple Authentication)* ermöglicht nur die Teilnehmerauthentifizierung. Sie basiert auf Passwörtern, die im Verzeichnis gespeichert werden. Die Teilnehmer authentifizieren sich gegenseitig durch unverschlüsselte Übermittlung ihres Namens und ihres Passwortes. Die einfache Authentifizierung sollte sich deshalb auf die lokale Kommunikation beschränken. Erst die *Starke Authentifizierung (Strong Authentication)* unterstützt neben der Authentifizierung von Teilnehmern auch den Schutz der übermittelten Information.

7 Ausblick

In diesem Artikel wird gezeigt, dass ein Verzeichnissystem gemäss X.500 ein komplexes, verteiltes System darstellt. Normengerechte Verwirklichungen werden deshalb entsprechend aufwendig sein. Trotzdem muss betont werden, dass X.500 nur ein Minimum der wünschbaren Konzepte und Funktionen umfasst. Im wesentlichen handelt es sich um:

- die Definition einer Struktur für einen hierarchischen Namensraum, der verschiedenen namensgebenden Instanzen Freiheit in bezug auf dessen konkrete Gestaltung lässt
- Richtlinien für eine sinnvolle Gestaltung dieses Namensraumes
- eine Spezifikation von Abfrage-, Such- und einfachen Änderungsoperationen sowie
- eine Vorschrift über die Zusammenarbeit von DSA, welche von verschiedenen Organisationen betrieben werden.

Zur Beantwortung der Frage, welche wünschbaren Funktionen in X.500 nicht definiert sind, lohnt es sich, einen Blick auf die Geschichte dieser Empfehlungen zu tun. Es kann festgestellt werden, dass in den Jahren 1984 bis 1986 Vorversionen diskutiert wurden, die zusätzlich zu den Funktionen der heutigen Version weitere Eigenschaften aufwiesen:

- Der vorgeschlagene Namensraum hatte nicht die Struktur eines Baumes, sondern eines Graphen ohne Zyklen. Damit konnten Objekte (auch ohne den Alias-Mechanismus) mehrere Namen haben. Der Begriff des *Distinguished Name* ist eigentlich ein Relikt aus diesen Entwürfen und wurde früher verwendet, um einen von mehreren möglichen Namen besonders auszuzeichnen.
- Mit dem Konzept von *deskriptiven Namen* versuchte man irgendwelche gespeicherte Information zur Benennung der Objekte zu verwenden. Ein deskriptiver Name für einen hypothetischen Mitbürger – als Menge von Attributen gegeben – könnte sein: *{Anzahl Haustiere = 3, Hobby = Segeln, Beruf = Elektriker}*. Es stellt sich natürlich sofort die Frage, ob diese Spezifikation überhaupt ein Name ist, das heisst, ob sie genau ein Objekt bezeichnet. Effiziente, verteilte Algorithmen, die diese Frage beantworten können, sind noch ein Thema der Forschung.

  Die Attraktivität von deskriptiven Namen besteht darin, dass sie den Unterschied zwischen White- und Yellow-Pages-Abfragen eliminieren.
- Unterstützung von replizierten Datenbeständen durch *Schattenkopien (Shadowing):* Ein DSA konnte einen Teil des Datenbestandes eines anderen DSA zum Zweck einer lokalen Speicherung und damit eines rascheren Zugriffes anfordern und gleichzeitig den Lieferanten der Information verpflichten, ihn über eventuelle Änderungen zu informieren.
- Änderungen beschränkten sich nicht auf das Entfernen oder das Erzeugen von Blattknoten, vielmehr konnte jeder Knoten des Graphen entfernt oder neu eingefügt werden. Man beachte, dass hier der Ausdruck DIT fehl am Platze wäre, da das Verzeichnis keine Baumstruktur aufwies.
- Ein Konzept für die Vergabe von Zugriffsrechten auf Einträgen, gestützt auf sogenannte *Zugriffskontrollisten (Access Control Lists).*

Die Vorversionen scheiterten denn auch vor allem wegen ihrer Komplexität. Es gab nicht nur begründete Zweifel an der Anwendbarkeit derartiger Normen, sondern auch Schwierigkeiten, eine korrekte Definition der Semantik von komplexen Operationen (z. B. des Entfernens eines Knotens innerhalb des Graphen) zu finden. Dabei war eines der Hauptprobleme die Sicherung der Konsistenz des Graphen. Es ist jedoch zu erwarten, dass mit den Erfahrungen bei der Entwicklung der ersten Anwendungen und dem Betrieb von Pilotsystemen einige der alten Ideen wieder aktuell werden und anlässlich einer Revision der Empfehlungen wieder diskutiert werden. Es ist zu erwarten, dass vor allem bezüglich der Zugriffskontrolle und der Verwaltung des Verzeichnisses Erweiterungen notwendig sein werden.

Den Autoren ist gegenwärtig eine kleine Zahl von Projekten bekannt, welche die Entwicklung von X.500-konformer Software zum Ziel haben. An der Hannover-Messe (CeBIT 89) wurden denn auch erste Prototypen

gezeigt. Es kann somit erwartet werden, dass bis Ende 1990 eine Anzahl experimenteller und kommerzieller Verzeichnissysteme interessierten Anwendern zur Verfügung stehen werden. Es sollte jedoch nicht erwartet werden, dass die Verfügbarkeit einer Anwendung allein schon zu einem korrekt funktionierenden, weltweiten Verzeichnis führen wird; vielmehr werden die Hauptprobleme im organisatorischen Bereich, das heisst im Zusammenspiel zwischen den Betreibern der einzelnen DSA, liegen.

Erste diesbezügliche Erfahrungen werden Pilotprojekte im universitären Rahmen, sowohl in Europa als auch in den USA, liefern. Zudem ist geplant, an wichtigen Messen (ähnlich wie bei der Einführung von X.400) sogenannte *Multivendor Shows* aufzubauen, in welchen X.500-Prototypen oder -Produkte verschiedener Hersteller gemeinsam ein Verzeichnissystem bilden.

Aus diesen Aktivitäten werden ebenfalls wertvolle Erkenntnisse betreffend das Testen von Implementationen in bezug auf ihre Normenkonformität gewonnen werden. Es ist anzunehmen, dass wie bei X.400 Testsuiten für X.500 entwickelt und normiert werden.

X.500 wird sich, trotz der hier genannten Einschränkungen, unzweifelhaft als ein wichtiges und unentbehrliches Bindeglied zwischen den verschiedenen OSI-Anwendungen erweisen.

Die nächste Nummer bringt unter anderem:
Vous pourrez lire dans le prochain numéro:

4/90

Geiser W.	Funknetzplanung für das Mobiltelefonsystem der Schweiz, Natel C Planification des réseaux de radiodiffusion pour le système suisse de téléphonie mobile Natel C
Schmid E. und Metzger K.	Natel C – Bau und Inbetriebnahme
Rothen A., Eggenberger W. und Müller J.	Erste Betriebserfahrungen mit Natel C und die weitere Planung Premières expériences d'exploitation avec le Natel C et planification future

---