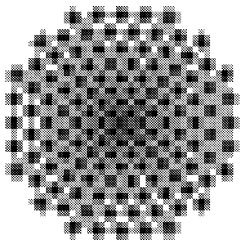
SEMINARARBEIT

X.500

im Rahmen des
Hauptseminars
Verzeichnisdienste



Universität Stuttgart
WS 2000/2001

Marc Bufé
Student der Softwaretechnik

INHALTSVERZEICHNIS

0. EINLEITUNG **1**

1. WOZU VERZEICHNISDIENSTE? **1**

2. DIE X.500 EMPFEHLUNGEN **1**

3. CHARAKTERISTIKA VON X.500 ..... **2**

4. INFORMATIONSMODELL VON X.500 ..... **2**

4.1 GRUNDLEGENDER AUFBAU 2
4.2 OBJEKTKLASSEN UND VERERBUNG 3
4.3 DIRECTORY INFORMATION TREE 3
4.4 ALIAS-KONZEPT 4
4.5 STRUKTURREGELN 4

5. OPERATIONSMODELL **4**

5.1 GRUNDLEGENDER AUFBAU 4
5.2 OPERATIONEN 4
5.4 METHODEN DER MODIFIKATION 5
5.5 METHODEN DER PARAMETERERGÄNZUNG 6

6. FUNKTIONSMODELL **6**

6.1 VERTEILUNG UND PARTITIONIERUNG 6
6.2 REFERENZEN ZWISCHEN DSAS 6
6.3 WEITERLEITUNG VON ANFRAGEN 7

7. REPLIKATION **7**

8. KOMMUNIKATIONSPROTOKOLLE **8**

9. SICHERHEIT **8**

10. ZUSAMMENFASSUNG **9**

0. Einleitung

Diese Ausarbeitung bezieht sich auf den gleichnamigen Vortrag im Rahmen des Hauptseminars "Verzeichnisdienste" der Fakultät Informatik an der Uni-Stuttgart im Wintersemester 2000/2001.

1. Wozu Verzeichnisdienste?

Verzeichnisdienste bieten dem Benutzer die Möglichkeit, über eine genormte Schnittstelle auf unterschiedliche Daten zu Objekten zuzugreifen. Die wahrscheinlich am häufigsten genutzte Form der Abfrage ist die sogenannte „white pages query“. Hierbei wird z.B. in einem Telefonbuch, über den Namen einer Person oder Institution auf den entsprechenden Datensatz zugegriffen, der dann Eigenschaften wie z.B. die Telefonnummer enthält. Eine andere Abfrageform ist die „yellow pages query“, hierbei sind lediglich die zu suchenden Eigenschaften eines Objektes bekannt – etwa vergleichbar mit den Gelben Seiten, in denen nach Branchen gesucht werden kann. Gegenüber Allzweckdatenbanken weisen Verzeichnisdienste jedoch einige Besonderheiten auf. So wird ein Eintrag in der Regel viel häufiger gelesen als geändert, man spricht von einem Verhältnis von 99 Lesezugriffen zu einem Schreibzugriff. Dafür wird hingenommen, dass Veränderungen an der Datenbasis nicht sofort global ausgeführt werden. Dies ermöglicht eine effizientere Implementierung gegenüber einer normalen Datenbank, die hierzu den betreffenden Datensatz erst komplett (insbesondere die Lesezugriffe) sperren müsste.

Nicht zuletzt dient ein Verzeichnisdienst dazu, Daten aus unterschiedlichen Quellen auf eine gemeinsame Struktur zu bringen, so dass nicht jede Applikation ihre eigene Datenbasis verwendet, was einen überflüssigen Mehraufwand bei der Pflege der Daten bedeuten würde. X.500 ist nun ein solcher Verzeichnisdienst, der die genannten Merkmale aufweist. Ziel bei X.500 war ein globales Verzeichnis, um Inkompatibilität und proprietäre Inseln weitgehend zu vermeiden. X.500 bezeichnet eine ganze Serie von Empfehlungen der ITU (International Telecommunication Union, einst CCITT), zu denen es auch entsprechende ISO-Standard (9594-1 bis 9594-9) gibt.

2. Die X.500 Empfehlungen

Die Serie der X.500 Empfehlungen gibt eine vollständige Beschreibung eines Verzeichnisdienstes als OSI Application Layer ab, der gesamte Standard umfasst mehrere hundert Seiten und gibt für alles bis ins Detail konkrete Implementierungsvorschläge. Hier eine kurze Übersicht:

- X.500 Konzepte (allgemeine Übersicht), X.511 Dienstspezifikation
- X.501, X.520, X.521 Informationsmodell – Siehe Kapitel 4
- X.518 Verteilung, X.525 Replikation – Siehe Kapitel 7
- X.519 Protokolle, X.509 Authentifizierung – Siehe Kapitel 8 und 9

3. Charakteristika von X.500

Ein X.500 Verzeichnisdienst wird als Client-Server Modell realisiert, d.h. die User verwenden eine Clientapplikation zum Zugriff auf den oder die Verzeichnisserver. Ein Client ist hierbei ein fertiges Programm, welches verwendet werden kann, um auf Informationen in einem X.500 Verzeichnisdienst zuzugreifen. Neben der Abfrage ist gegebenenfalls auch eine Manipulation der Inhalte und/oder der Administration des Dienstes möglich.

Die Datenbasis ist hierarchisch als Baum organisiert, wobei der Standard die Erweiterung durch andere hierarchische Organisationsstrukturen offen lässt. Typischerweise beinhalten X.500 Verzeichnisse eine relativ große Datenmenge, die sich oft auch auf weltweit verteilte Objekte bezieht. Daher sieht der X.500 Standard eine Verteilung der Datenbasis auf mehrere physikalisch getrennte Server vor. Diese befinden sich typischerweise dort, wo die jeweiligen Objekte angesiedelt sind. Dies ermöglicht zudem die Administration der Daten vor Ort.

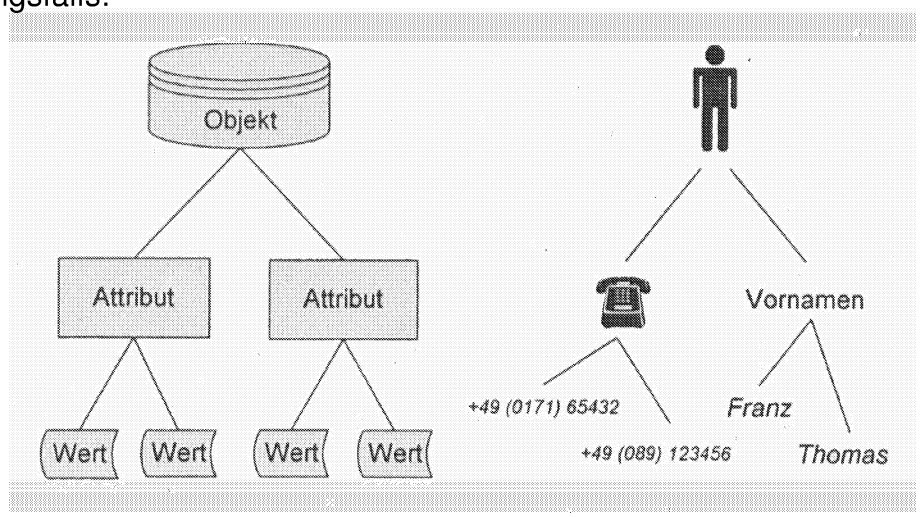
Im Gegensatz zu z.B. Domain Name System (DNS) definiert X.500 verschiedene Möglichkeiten zur Authentifizierung (bspw. mit X.509), die auch auf starker Kryptographie (RSA) basieren, und damit eine Nutzung auch in nicht vertrauenswürdiger Umgebung ermöglichen.

X.500 eignet sich zur Verwaltung elektronischer Daten aller Art, was seinen universell einsetzbaren Charakter unterstreicht.

4. Informationsmodell von X.500

4.1 Grundlegender Aufbau

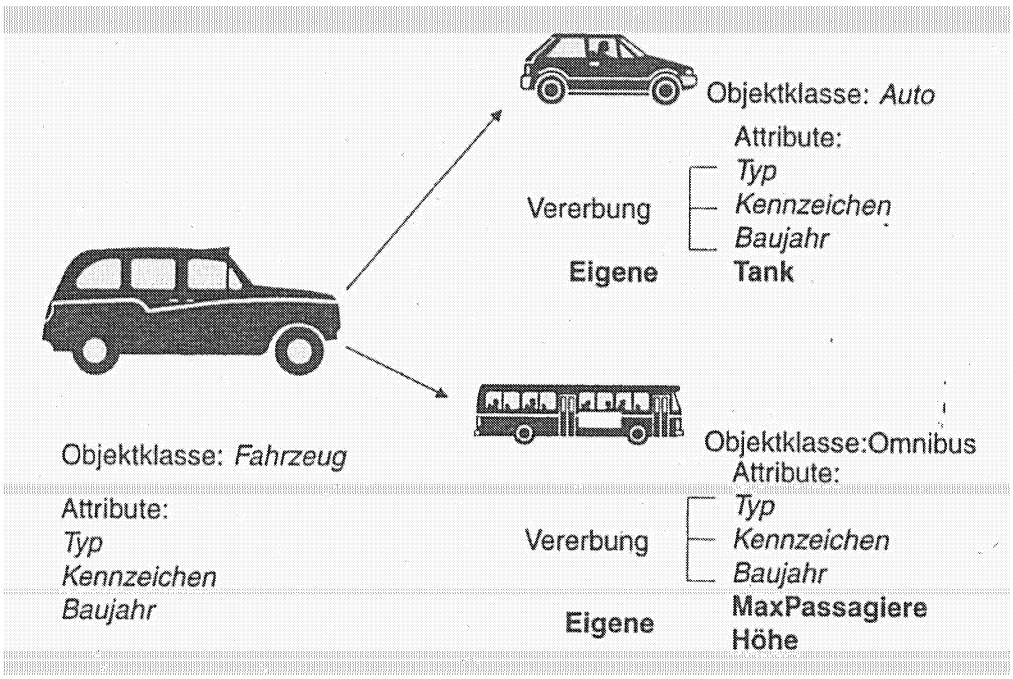
Die Daten der zu verwaltenden Objekte werden in Form von Einträgen gespeichert, die jeweils Attribute mit einem oder mehreren Attributswerten enthalten. Die Attribute sind dabei in Attributsklassen unterteilt, wobei eine solche den für das jeweilige Attribut zur Verfügung stehenden Wertebereich festlegt. Hier eine Illustration eines Anwendungsfalls:



4.2 Objektklassen und Vererbung

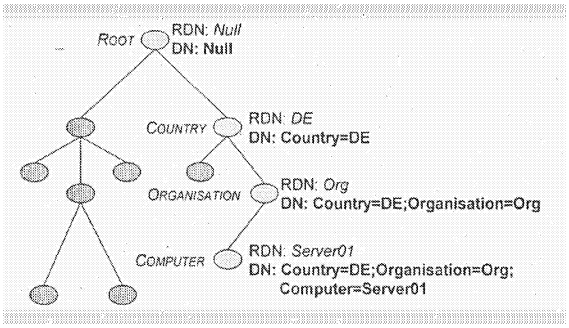
Um festzulegen, über welche Attribute ein Eintrag verfügt, sind die in der Datenbasis enthaltenen Objekte in Objektklassen unterteilt.

Diese sind ihrerseits hierarchisch organisiert, wobei eine Objektklasse, die Subklasse einer anderen ist, alle Attribute erbt und eigene hinzufügen kann. So könnte z.B. jedes Fahrzeugs eines Fuhrunternehmens ein Attribut Baujahr besitzen, wobei Busse als Unterklasse der Fahrzeuge zusätzlich über ein Attribut MaxPassagiere bzw. AnzahlSitzplätze verfügen könnten.



4.3 Directory Information Tree

Die einzelnen Einträge werden nun als Baum organisiert, wodurch der Directory Information Tree (DIT) entsteht.
Jedes Objekt hat ein speziell ausgezeichnetes Attribut, das den Relatively Distinguished Name (RDN) bestimmt, der innerhalb einer Unterebene eines Knotens eindeutig sein muss.



Dadurch entstehen auf das ganze Verzeichnis global gesehen eindeutige Namen aus der Aneinanderreihung der RDNs aller Knoten von der Wurzel abwärts bis zu einem konkreten Eintrag. Diese globalen, also tatsächlich weltweit eindeutigen, Namen sind die Distinguished Names (DN). Ein direkter Zugriff auf ein Objekt ist über den DN möglich.

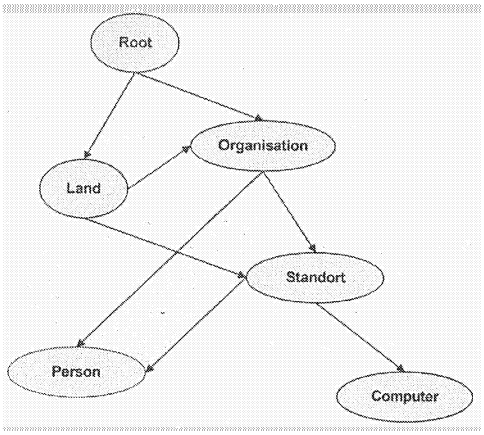
Einer Sonderrolle kommt der Wurzel zu. Diese ist nur aus Vollständigkeitsgründen vorhanden, eigentlich namenlos, und besitzt immer den RDN „Null“.

4.4 Alias-Konzept

Im Directory Information Tree wird auch ein Alias-Konzept unterstützt. Dieses wird dadurch realisiert dass ein Objekt lediglich einen Verweis auf ein anderes Objekt besitzen kann. Vorteilhaft ist dies um eine bestimmte Information nicht mehrfach warten zu müssen. Dies erspart Kosten, ist somit effizienter, und angesichts der Redundanz auch effektiver.

4.5 Strukturregeln

Strukturregeln im Directory Information Tree sollen sicherstellen, dass Objekte nur bestimmten anderen untergeordnet werden können. Dies verhindert bspw. dass eine Person einem Auto oder eine Unternehmung einem Computer untergeordnet wird.



In der Abbildung ist durch Pfeile angedeutet wer wem untergeordnet ist. So kann hier eine Person nur einer Organisation oder einem Standort untergeordnet sein, ein Computer sogar nur einem Standort.

5. Operationsmodell

5.1 Grundlegender Aufbau

Wie anfangs erwähnt, basiert der Zugriff durch die Benutzer auf einem Client-Server Modell. Die Benutzer, welche nicht zwingend reale Benutzer, sondern z.B. auch Applikationen sein können, verwenden dabei den sogenannten Directory User Agent (DUA). Die Schnittstelle über den der Benutzer bzw. eine Applikation mit dem DUA kommuniziert ist nicht festgelegt. Daher ist die Einbettung eines DUAs in eine Applikation zulässig. Es muß also nicht unbedingt ein separater Client eingesetzt werden.

5.2 Operationen

Das Directory bietet dem Benutzer verschiedene Services an, die sich in die Gruppe der Abfrage (Interrogation), der Veränderung (Modification) und der Parameter-ergänzung unterteilen lassen. Generell werden alle Methoden der Services asynchron abgearbeitet.

5.3 Methoden der Abfrage

Die Gruppe der Abfrage umfasst folgende Methoden:

- READ
  Das Lesen eines Eintrages und zugehöriger Attribute über den jeweiligen DN. Entspricht weitgehend der eingangs erwähnten „white pages query“, also dem direkten Zugriff auf ein bereits bekanntes Objekt um dessen Attribute auslesen zu können.

- COMPARE
  Das Vergleichen von Attributen eines Eintrages mit vorgegebenen Werten, damit kann bspw. Ein Passwort verifiziert werden, ohne dass der Benutzer die Möglichkeit hat dieses auszulesen.

- LIST
  Auflisten aller Unterknoten eines vorgegebenen Knotens des DIT

- SEARCH
  Die Suche nach vorgegebenen Attributswerten in einem Teilbaum des DIT. Entspricht weitgehend der eingangs erwähnten „yellow pages query“, bspw. die Suche nach allen Schulen in einer Stadt.

- ABANDON
  Teilt dem Verzeichnis mit, dass das Ergebnis einer bereits abgesetzten Anfrage nicht mehr benötigt wird. Diese Operation wird benötigt, da eine Anfrage asynchron abgearbeitet wird.

5.4 Methoden der Modifikation

Die Gruppe der Modifikation umfasst folgende Methoden:

- ADD ENTRY
  Hinzufügen eines Eintrags

- REMOVE ENTRY
  Löschen eines Eintrags

- MODIFY ENTRY
  Verändern und Hinzufügen von Attributen zu einem vorhandenen Eintrag, das Ändern des RDNs wird jedoch nicht unterstützt.

- MODIFY RDN
  Ändern des RDNs, ermöglicht auch das Verschieben eines Knotens des DIT mit allen Unterknoten an eine andere Stelle.

5.5 Methoden der Parameterergänzung

Die Gruppe der Parameterergänzung umfasst folgende Methoden:

- SERVICE CONTROLS
  Limits auf Bearbeitungszeit, Größe und Interaktionsmodelle

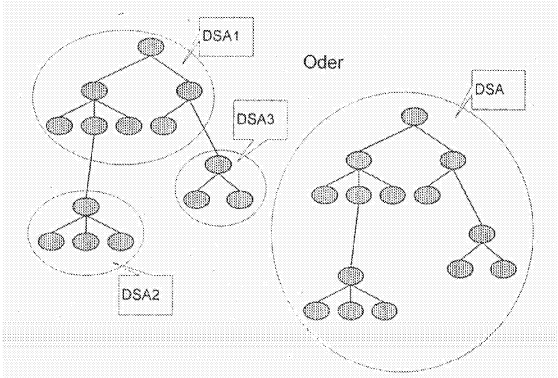
- SECURITY
  Verlangen von Signaturen, zum Zweck der Authentifizierung

- FILTERS
  Einschränkungen des Ergebnisses einer Anfrage durch verschiedene Filterkriterien

6. Funktionsmodell

6.1 Verteilung und Partitionierung

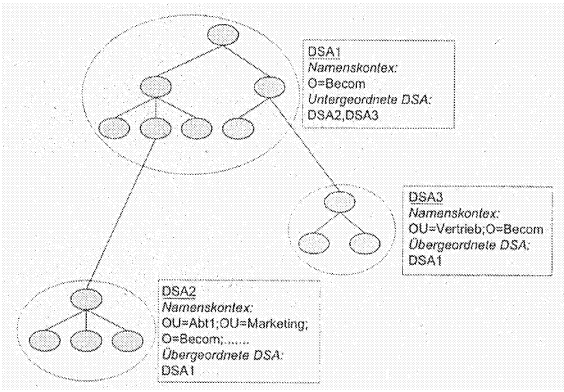
Das Verzeichnis ist typischerweise nicht zentral an einem Ort gespeichert, sondern die Information ist auf mehrere sogenannte Directory System Agents (DSA), verteilt. Diese kommunizieren untereinander und dienen den DUAs als Zugangspunkte (den sogenannten Access Points). Ein DUA ist dabei nicht auf einen bestimmten DSA festgelegt, sondern kann bei Bedarf auch auf mehrere DSAs zugreifen.



Die Aufteilung der Informationsbasis auf mehrere DSAs entspricht einer Partitionierung des DITs, wobei ein DSA auch mehrere Partitionen verwalten kann (siehe Bild).

6.2 Referenzen zwischen DSAs

Um ein konkretes Funktionieren des Verzeichnisses zu gewährleisten, müssen die DSAs deshalb ein gewisses Wissen über die Partitionierung des DIT besitzen. Selbiges wird in Referenzen gespeichert.



Zu jeder Partition ist gespeichert, welcher DSA die im DIT darüber liegende Partition speichert (superior reference), welche DSAs die untergeordneten Partitionen speichern (subordinate references), sowie zur Performance-Steigerung und mangels einer Wurzelpartition auf der obersten (First Level) Ebene Querverweise (cross references).

6.3 Weiterleitung von Anfragen

Anhand des Wissens über die Verteilung des Verzeichnisses hat ein DSA mehrere Möglichkeiten, eine Anfrage zu Daten, welche er selbst nicht vorhält, zu bearbeiten:

- REFERAL
  Der DSA gibt dem Anfragesteller (einem anderen DSA oder auch dem DUA) als Antwort einen anderen DSA, an den er sich wenden soll. Nachteil: Der DUA kann mitunter erkennen, wie die Information verteilt ist – was aber zugelassen werden muss.

- CHAINING
  Der DSA fragt seinerseits einen anderen DSA und reicht das Ergebnis an den Anfragesteller zurück. Nachteil: Das Ergebnis einer Anfrage muss über alle DSAs hindurch zurückgeleitet werden, was einen längeren Rückweg bedeutet.

- MULTICASTING
  Kann der DSA nicht entscheiden, welcher von mehreren möglichen DSAs der beste für eine Anfrage ist, kann er sie auch als Multicast an mehrere gleichzeitig schicken.

Alle drei Methoden können im Verlauf der Beantwortung einer Anfrage in beliebiger Kombination vorkommen, da jeder DSA von neuem entscheiden kann, wie er die ihm gestellte Anfrage bearbeitet.

7. Replikation

Prinzipiell finden zwei Arten der Replikation Anwendung, das Caching und das Shadowing. Caching wird in X.500 nicht näher spezifiziert. Shadowing ist das Kopieren von Informationen eines anderen DSA und erfolgt aufgrund sogenannter „Shadowing Agreements“ zwischen jeweils zwei DSAs.

Die Replikation hat zwei wesentliche Ziele: zum einen die Verbesserung der Performance, indem häufig benötigte Informationen andere Teile des Directory lokal verfügbar gemacht werden, sowie eine Erhöhung der Verfügbarkeit durch die damit verbundene redundante Speicherung der Information.

Natürlich müssen auch die mit dem Shadowing verbundenen Probleme gelöst werden. So müssen z.B. DSAs mit im Zugriff beschränkter Information darauf achten, dass entweder alle zur Verifikation der Zugriffsberechtigung nötigen Informationen dem kopierenden DSA zur Verfügung stehen, oder solche Informationen nicht repliziert werden. Modifikationen werden grundsätzlich nur am Master DSA, also dem DSA der die originalen Informationen vorhält, durchgeführt. Zur Wahrung der Konsistenz stehen zudem Möglichkeiten zur Verfügung, über die ein DSA den ihn replizierenden DSAs mitteilen kann, dass neue Information vorliegt, bzw. diese auch direkt weitergeben (mittels des push-Verfahrens).

8. Kommunikationsprotokolle

Die X.500 Kommunikationsprotokolle sind als OSI Application (Layer 7) definiert, die zur Kommunikation verwendeten Application-Service-Elemente sind ACSE (Association Control Service Element, für Verbindungsab- und aufbau) und ROSE (Remote Operations Service Element, für die eigentliche Kommunikation) sowie optional das RTSE (Reliable Transfer Service Element).

Definiert werden die Protokolle zur Kommunikation zwischen den DSAs (DSP, Directory System Protocol) sowie zwischen den DSAs und DUAs (DAP, Directory Access Protocol), die Kommunikation zwischen dem User und DUA oder auch einer anderen Applikation bleibt hingegen undefiniert.

Es ergeben sich also folgende Protokolle im Einsatz:

- DAP – Directory Access Protocol
- DSP – Directory System Protocol
- DOP – Directory Operational Binding Management Protocol
- DISP – Directory Information Shadowing Protocol

9. Sicherheit

Mit X.509 stehen dem X.500 Directory Möglichkeiten der Kryptographie zur Verfügung. Über RSA und Zertifikate für öffentliche Schlüssel können sich sowohl Benutzer gegenüber dem Directory, als auch das Directory sich gegenüber dem Benutzer sicher authentifizieren und hiermit die Integrität der Daten, die ja u.U. von einem entfernteren DSA stammen, sichergestellt werden. X.509 findet teilweise auch ausserhalb von X.500 Anwendung, z.B. im PEM-Standard für sichere eMail.

Außerdem bietet X.500 vielfältige Möglichkeiten des Access Controls, also der Kontrolle darüber, wer auf welche Daten zugreifen kann. Es kann sowohl der Zugriff auf Einträge als auch auf einzelne Attribute beschränkt werden sowie festgelegt werden, welche Arten des Zugriffs (read, browse, modify) bestehen.

10. Zusammenfassung

Die X.500 Standards geben in der aktuellen Fassung eine vollständige Definition eines universell einsetzbaren Verzeichnisdienstes ab, mit Verteilung, Redundanz, Verschlüsselung und Authentifizierung sowie sehr flexiblen Möglichkeiten der Abfragebearbeitung. Durch die Verteilung der Information (und damit meist auch der Administration) ist die Verwaltung einer erheblich größeren Zahl von Nutzern möglich, als dies bspw. beim ursprünglich auf 70000 Einträge limitierten whois-Dienst der Fall war.

Vorteilhaft ist, dass X.500 eine unternehmensähnliche hierarchische Struktur aufweist und damit ein Unternehmen leicht abgebildet werden kann. Da jeder Verwalter nur für seine Ebene zuständig ist, kann schnell und vor Ort auf Veränderungen reagiert werden, was der gewünschten Dezentralisierung der Information entspricht. In der Praxis wird X.500 nur zur Verwaltung von Personendaten und eMail-Adressen verwendet, obgleich es auch alles andere wie z.B. Bilder speichern könnte.

Nachteilhaft ist, dass eine Implementierung eines X.500 Verzeichnisses gerade wegen der Verwendung des OSI Stacks relativ aufwendig ist, und auch die Geschwindigkeit der Abfragebearbeitung hinter neueren Verzeichnisdiensten wie bspw. LDAP zurückbleibt.