The drawings contained in this Recommendation have been done in AUTOCAD

Recommendation X.500

THE DIRECTORY - OVERVIEW OF CONCEPTS, MODELS AND SERVICES [1)]

(Melbourne, 1988)

CONTENTS

0 *Introduction*

1 *Scope and field of application*

2 *References*

3 *Definitions*

- 3.1 OSI reference model definitions
- 3.2 Basic directory definitions
- 3.3 Directory model definitions
- 3.4 Distributed operation definitions

4 *Abbreviations*

5 *Overview of the directory*

6 *The directory information base (DIB)*

7 *The directory service*

- 7.1 Introduction
- 7.2 Service qualification
- 7.3 Directory interrogation
- 7.4 Directory modification
- 7.5 Other outcomes

8 *The distributed directory*

- 8.1 Functional model
- 8.2 Organizational model
- 8.3 Operation of the model

9 *Directory protocols*

Annex A - Applying the directory

- A.1 The directory environment
- A.2 Directory service characteristics
- A.3 Patterns of use of the directory
- A.4 Generic applications

---

1) Recommendation X.500 and ISO 9594-1, The Directory - Overview of Concepts, Models and Services, were developed in close collaboration and are technically aligned.

0 Introduction

0.1 This document, together with the others of the series, has been produced to facilitate the interconnection of information processing systems to provide directory services. The set of all such systems, together with the directory information which they hold, can be viewed as an integrated whole, called the Directory. The information held by the Directory, collectively known as the Directory Information Base (DIB), is typically used to facilitate communication between, with or about objects such as application entities, people, terminals and distribution lists.

0.2 The Directory plays a significant role in Open Systems Interconnection, whose aim is to allow, with a minimum of technical agreement outside of the interconnection standards themselves, the interconnection of information processing systems:

- from different manufacturers;
- under different managements;
- of different levels of complexity; and
- of different ages.

0.3 This Recommendation introduces and models the concepts of the Directory and of the DIB and overviews the services and capabilities which they provide. Other Recommendations make use of these models in defining the abstract service provided by the Directory, and in specifying the protocols through which this service can be obtained or propagated.

1 Scope and field of application

1.1 The Directory provides the directory capabilities required by OSI applications, OSI management processes, other OSI layer entities, and telecommunication services. Among the capabilities which it provides are those of "user-friendly naming" whereby objects can be referred to by names which are suitable for citing by human users (though not all objects need have user-friendly names); and "name- to-address mapping" which allows the binding between objects and their locations to be dynamic. The latter capability allows OSI networks, for example, to be "self-configuring" in the sense that addition, removal and the changes of object location do not affect OSI network operation.

1.2 The Directory is not intended to be a general-purpose data base system, although it may be built on such systems. It is assumed, for instance, that, as is typical with communications directories, there is a considerably higher frequency of "queries" than of updates. The rate of updates is expected to be governed by the dynamics of people and organizations, rather than, for example, the dynamics of networks. There is also no need for instantaneous global commitment of updates: transient conditions where both old and new versions of the same information are available, are quite acceptable.

1.3 It is a characteristic of the Directory that, except as a consequence of differing access rights or unpropagated updates, the results of directory queries will not be dependent on the identity or location of the enquirer. This characteristic renders the Directory unsuitable for some telecommunications applications, for example some types of routing.

2 References

Recommendation X.200 -Open Systems Interconnection - Basic Reference Model.

Recommendation X.208 -Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).

Recommendation X.501 - The Directory - Models.

Recommendation X.509 - The Directory - Authentication framework.

Recommendation X.511 - The Directory - Abstract Service Definition.

Recommendation X.518 - The Directory - Procedures for Distributed Operation.

Recommendation X.519 - The Directory - Protocol Specifications.

Recommendation X.520 - The Directory - Selected Attribute Types.

Recommendation X.521 - The Directory - Selected Object Classes.

Recommendation X.219 - Remote Operations - Model, Notation and Service Definition.

Recommendation X.229 - Remote Operations - Protocol Specification.

3 Definitions

The definitions contained in this make use of the abbreviations defined in § 4.

3.1 *OSI reference model definitions*

This Recommendation is based on the concepts developed in Recommendation X.200, and makes use of the following terms therein defined:

a) application-entity;

b) Application Layer;

c) application process;

d) application protocol data unit;

e) application service element.

3.2 *Basic directory definitions*

a) The Directory: a collection of open systems cooperating to provide directory services;

b) Directory Information Base (DIB): the set of information managed by the Directory;

c) (Directory) user: the end user of the Directory, i.e. the entity or person which accesses the Directory.

3.3 *Directory model definitions*

This Recommendation makes use of the following terms defined in Recommendation X.501.

a) Administration Directory Management Domain;

b) alias;

c) attribute;

d) attribute type;

e) attribute value;

f) Directory Information Tree (DIT);

g) Directory Management Domain (DMD);

h) Directory System Agent (DSA);

i) Directory User Agent (DUA);

j) distinguished name;

k) entry;

l) name;

m) object (of interest);

n) Private Directory Management Domain;

o) relative distinguished name;

p) root;

q) schema;

r) subordinate object;

s) superior entry;

t) superior object;

u) tree.

3.4 *Distributed operation definitions*

This Recommendation makes use of the following terms defined in Recommendation X.518:

a) chaining;

b) multicasting;

c) referral.

4 Abbreviations

ADDMD Administration Directory Management Domain

DAP Directory Access Protocol

DIB Directory Information Base

DIT Directory Information Tree

DMD Directory Management Domain

DSA Directory System Agent

DSP Directory System Protocol

DUA Directory User Agent

OSI Open Systems Interconnection

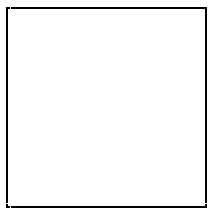
PRDMD Private Directory Management Domain

RDN Relative Distinguished Name

5 Overview of the Directory

5.1 The Directory is a collection of open systems which cooperate to hold a logical data base of information about a set of objects in the real world. The users of the Directory, including people and computer programs, can read or modify the information, or parts of it, subject to having permission to do so. Each user is represented in accessing the Directory by a Directory User Agent (DUA), which is considered to be an application-process. These concepts are illustrated in Figure 1/X.500.

FIGURE 1/X.500 - -T0704210-88



Note - This series of Recommendations refers to the Directory in the singular, and reflects the intention to create, through a single, unified, name space, one logical directory composed of many systems and serving many applications. Whether or not these systems choose to interwork will depend on the needs of the applications they support. Applications dealing with non-intersecting worlds of objects, may have no such need. The single name space facilitates later interworking should the needs change.

5.2 The information held in the Directory is collectively known as the Directory Information Base (DIB). Clause 6 of this Recommendation overviews its structure.

5.3 The Directory provides a well-defined set of access capabilities, known as the abstract service of the Directory, to its users. This service, which is overviewed in 7 of this Recommendation provides a simple modification and retrieval capability. This can be built on with local DUA functions to provide the capabilities required by the end-users.

5.4 It is likely that the Directory will be distributed, perhaps widely distributed, both along functional and organizational lines. 8 overviews the corresponding models of the Directory. These have been developed in order to provide a framework for the cooperation of the various components to provide an integrated whole.

5.5 The provision and consumption of the Directory services requires that the users (actually the DUAs) and the various functional components of the Directory should cooperate with one another. In many cases this will require cooperation between application processes in different open systems, which in turn requires standardized application protocols, overviewed in 9, to govern this cooperation.

5.6 The Directory has been designed so as to support multiple applications, drawn from a wide range of possibilities. The nature of the application supported will govern which objects are listed in the Directory, which users will access the information, and which kinds of access they will carry out. Applications may be very specific, such as the provision of distribution lists for electronic mail, or generic, such as the "inter-personal communications directory" application. The Directory provides the opportunity to exploit commonalities among the applications:

- a single object may be relevant to more than one application; perhaps even the same piece of information about the same object may be so relevant.

To support this, a number of object classes and attribute types are defined, which will be useful across a range of applications. These definitions are contained in Recommendations X.520 and X.521:

- certain patterns of use of the Directory will be common across a range of applications: this area is overviewed further in Annex A.

6 The Directory Information Base (DIB)

Note - The DIB, and its structure, are defined in Recommendation X.501.

6.1 The DIB is made up of information about objects. It is composed of (directory) entries, each of which consists of a collection of information on one object. Each entry is made up of attributes, each with a type and one or more values. The types of attribute which are present in a particular entry are dependent on the class of object which the entry describes.

6.2 The entries of the DIB are arranged in the form of a tree, the Directory Information Tree (DIT) where the vertices represent the entries. Entries higher in the tree (nearer the root) will often represent objects such as countries or organizations while entries lower in the tree will represent people or application processes.

Note - The services defined in this Recommendation operate only on a tree-structured DIT. This Recommendation does not preclude the existence in the future of other structures (as the need arises).

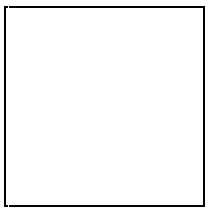
6.3 Every entry has a distinguished name, which uniquely and unambiguously identifies the entry. These properties of the distinguished name are derived from the tree structure of the information. The distinguished name of an entry is made up of the distinguished name of its superior entry, together with specially nominated attribute values (the distinguished values) from the entry.

6.4 Some of the entries at the leaves of the tree are alias entries, while all other entries are object entries. Alias entries point to object entries, and provide the basis for alternative names for the corresponding objects.

6.5 The Directory enforces a set of rules to ensure that the DIB remains well-formed in the face of modifications over time. These rules, known as the Directory schema, prevent entries having the wrong types of attributes for its object class, attribute values being of the wrong form for the attribute type, and even entries having subordinate entries of the wrong class.

6.6 Figure 2/X.500 illustrates the above concepts of the DIT and its components.

FIGURE 2/X.500 - T0704220-88



6.7 Figure 3/X.500 gives a hypothetical example of a DIT. The tree provides examples of some of the types of attributes used to identify different objects. For example the name:

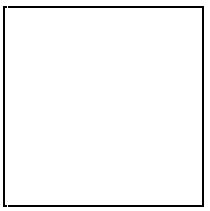
{C = GB, L = Winslow, O = Graphic Services, CN = Laser Printer}

identifies the application entity "Laser Printer" which has in its distinguished name the geographical attribute of Locality. The residential person John Jones, whose name is GB

{C = GB, L = Winslow, CN = John Jones}

has the same geographical attribute in his distinguished name.

FIGURE 3/X.500 - T0704230-88



6.8 The growth and form of the DIT, the definition of the Directory schema, and the selection of distinguished names for entries as they are added, is the responsibility of various authorities, whose hierarchical relationship is reflected in the shape of the tree. The authorities must ensure, for example, that all of the entries in their jurisdiction have unambiguous distinguished names, by carefully managing the attribute types and values which appear in those names. Responsibility is passed down the tree from superior to subordinate authorities, with control being exercised by means of the schema.

7 The Directory service

Note - The definition of the abstract service of the Directory can be found in Recommendation X.511.

7.1 *Introduction*

7.1.1 This provides an overview of the service provided to users, as represented by their DUAs, by the Directory. All services are provided by the Directory in response to requests from DUAs. There are requests which allow interrogation of the Directory, as described in § 7.3, and those for modification, as described in 7.4. In addition, requests for service can be qualified, as described in § 7.2. The Directory always reports the outcome of each request that is made of it. The form of the normal outcome is specific to the request, and is evident from the description of the request. Most abnormal outcomes are common to several requests. The possibilities are described in 7.5.

7.1.2 A number of aspects of the eventual directory service are not presently provided by the standards specified in this series of Recommendations. The corresponding capabilities will, therefore, need to be provided as a local function until such time as a standardized solution is available. These capabilities include:

- addition and deletion of arbitrary entries, thus allowing a distributed Directory to be created;
- the management of access control (i.e. granting or withdrawing permission for a particular user to carry out a particular access on a particular piece of information);
- the management of the Directory schema;
- the management of knowledge information;
- the replication of parts of the DIB.

Note - This list is not necessarily exhaustive.

7.1.3 The Directory ensures that changes to the DIB, whether the result of a Directory service request, or by some other (local) means, result in a DIB which continues to obey the rules of the Directory schema.

7.1.4 A User and the Directory are bound together for a period of time at an access point to the Directory. At the time of binding, the User and the Directory optionally verify each other's identity.

7.2 *Service qualification*

7.2.1 *Service controls*

A number of controls can be applied to the various service requests, primarily to allow the user to impose limits on the use of resources which the Directory must not surpass. Controls are provided on, among other things: the amount of time, the size of the results, the scope of search the interaction modes, and on the priority of the request.

7.2.2 *Security parameters*

Each request may be accompanied by information in support of security mechanisms for protecting the Directory information. Such information may include the user's request for various kinds of protection; a digital signature of the request, together with information to assist the correct party to verify the signature.

7.2.3 *Filters*

A number of requests whose outcome involves information from or concerning a number of entries, may carry with them a filter. A filter expresses one or more conditions that an entry must satisfy in order to be returned as part of the outcome. This allows the set of entries returned to be reduced to only those relevant.

7.3 *Directory interrogation*

7.3.1 *Read*

A read request is aimed at a particular entry, and causes the values of some or all of the attributes of that entry to be returned. Where only some attributes are to be returned, the DUA supplies the list of attribute types of interest.

7.3.2 *Compare*

A compare request is aimed at a particular attribute of a particular entry, and causes the Directory to check whether a supplied value matches a value of that attribute.

Note - For example, this can be used to carry out password checking, where the password, held in the Directory, might be inaccessible for read, but accessible for compare.

7.3.3 *List*

A list request causes the Directory to return the list of immediate subordinates of a particular named entry in the DIT.

7.3.4 *Search*

A search request causes the Directory to return information from all of the entries within a certain portion of the DIT which satisfy some filter. The information returned from each entry consists of some or all of the attributes of that entry, as with read.

7.3.5 *Abandon*

An abandon request, as applied to an outstanding interrogation request, informs the Directory that the originator of the request is no longer interested in the request being carried out. The Directory may, for example, cease processing the request, and may discard any results so far achieved.

7.4 *Directory modification*

7.4.1 *Add entry*

An add entry request causes a new leaf entry (either an object entry, or an alias entry) to be added to the DIT.

Note - In its present form this service is intended to be used to add entries which will remain as leaves, such as entries for people or application entities, rather than to add whole subtrees by repeated applications of this service. It is envisaged that the service will be enhanced in the future to cater to the more general case.

7.4.2 *Remove entry*

A remove entry request causes a leaf entry to be removed from the DIT.

Note - As with add entry, this service is presently intended for operation on "true leaf" entries, and will be enhanced in the future for the general case.

7.4.3 *Modify entry*

A modify entry request causes the Directory to execute a sequence of changes to a particular entry. Either all of the changes are made, or none of them, and the DIB is always left in a state consistent with the schema. The changes allowed include the addition, removal, or replacement of attributes or attribute values.

7.4.4 *Modify relative distinguished name*

A modify relative distinguished name (RDN) request causes the relative distinguished name of a leaf entry (either an object entry or an alias entry) in the DIT to be modified by the nomination of different distinguished attribute values.

7.5 *Other outcomes*

7.5.1 *Errors*

Any service may fail, for example because of problems with the user supplied parameters, in which case an error is reported. Information is returned with the error, where possible, to assist in correcting the problem. However, in general, only the first error encountered by the Directory is reported. Besides the above-mentioned example of problems with the parameters supplied by the user (particularly invalid names for entries or invalid attribute types), errors may arise from violations of security policy, schema rules, and service controls.

7.5.2 *Referrals*

A service may fail because the particular access point to which the DUA is bound is not the most suitable for carrying out the request, e.g. because the information affected by the request is (logically) far away from the access point. In this case the Directory may return a referral, which suggests an alternative access point at which the DUA can make its request.

Note - The Directory and the DUA may each have a preference as to whether referrals are used, or whether the requests are chained (see § 8.3.3.2). The DUA can express its preference by means of service controls. The Directory makes the final decision as to which approach is used.

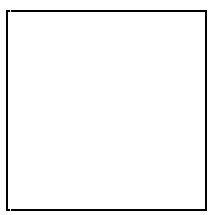
8 The distributed Directory

Note - the models of the directory are defined in Recommendation X.501 while the procedures for the operation of the distributed Directory are specified in Recommendation X.518.

8.1 *Functional model*

The functional model of the Directory is shown in Figure 4/X.500.

FIGURE 4/X.500 - T0704240-88



A Directory System Agent (DSA) is an OSI application process which is part of the Directory and whose role is to provide access to the DIB to DUAs and/or other DSAs. A DSA may use information stored in its local data base or interact with other DSAs to carry out requests. Alternatively, the DSA may direct a requestor to another DSA which can help carry out the request. Local data bases are entirely implementation dependent.

8.2 *Organizational model*

8.2.1 A set of one or more DSAs and zero or more DUAs managed by a single organization may form a Directory Management Domain (DMD). The organization concerned may or may not elect to make use of this series of Recommendations to govern the communications among the functional components within the DMD.

8.2.2 Subsequent Recommendations specify certain aspects of the behaviour of DSAs. For this purpose, a group of DSAs within one DMD may, at the option of the organization which manages the DMD, behave as a single DSA.

8.2.3 A DMD may be an Administration DMD (ADDMD), or a Private DMD (PRDMD), depending on whether or not it is being operated by a public telecommunications organization.

Note - It should be recognized that the provision of support for private directory systems by CCITT members falls within the framework of national regulations. Thus, the technical possibilities described may or may not be offered by an Administration which provides directory services. The internal operation and configuration of private DMDs is not within the scope of envisaged CCITT Recommendations.

8.3 *Operation of the model*

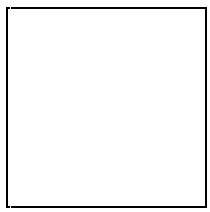
8.3.1 The DUA interacts with the Directory by communicating with one or more DSAs. A DUA need not be bound to any particular DSA. It may interact directly with various DSAs to make requests. For some administrative reasons, it may not always be possible to interact directly with the DSA which needs to carry out the request, e.g. to return some directory information. It is also possible that the DUA can access the Directory through a single DSA. For this purpose, DSAs will need to interact with each other.

8.3.2 The DSA is concerned with carrying out the requests of DUAs and with obtaining the information where it does not have the necessary information. It may take the responsibility to obtain the information by interacting with other DSAs on behalf of the DUA.

8.3.3 A number of cases of request handling have been identified, as illustrated in Figures 57/X.500, and described below.

8.3.3.1 In Figure 5a/X.500, the DSA C receives a referral from DSA A and is responsible for either conveying the request to the DSA B (named in the referral from DSA A) or conveying the referral back to the originating DUA.

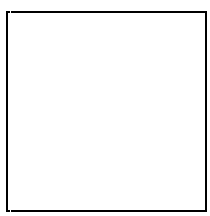
FIGURE 5a/X.500 - T0704250-88



Note - If DSA C returns the referral to the DUA, the "request (to B)" will not occur. Similarly, if DSA C conveys the request to DSA B, it will not return a referral to the DUA.

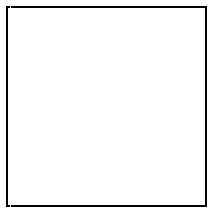
In Figure 5b/X.500, the DUA receives the referral from DSA C and is responsible for reissuing the request directly to DSA A (named in the referral from DSA C).

FIGURE 5b/X.500 - 0704260-88



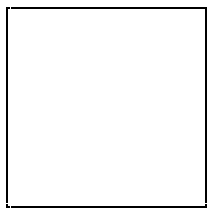
8.3.3.2 Figure 6/X.500 shows DSA chaining, whereby the request can be passed through several DSAs before the response is returned.

FIGURE 6/X.500 - T0704270-88



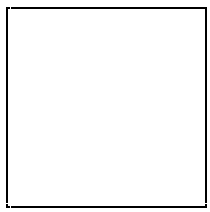
8.3.3.3 Figure 7/X.500 shows multicasting, where the DSA associated with the DUA carries out the request by forwarding it to two or more other DSAs, the request to each DSA being identical.

FIGURE 7/X.500 - T0704280-88



8.3.4 All of the approaches have their merits. For example, the approach in Figure 5/X.500 may be used where it is desirable to offload the burden from the local DSA. In other circumstances, a hybrid approach that combines a more elaborate set of functional interactions may be needed to satisfy the initiator's request, as illustrated in Figure 8/X.500.

FIGURE 8/X.500 - T0704290-88



9 Directory protocols

Note - The OSI application layer protocols defined to allow DUAs and DSAs in different open systems to cooperate are specified in Recommendation X.519.

9.1 There are two Directory protocols:

- the Directory Access Protocol (DAP), which defines the exchange of requests and outcomes between a DUA and a DSA;
- the Directory System Protocol (DSP), which defines the exchange of requests and outcomes between two DSAs.

9.2 Each protocol is defined by an application context, each containing a set of protocol elements. For example, the DAP contains protocol elements associated with interrogating and modifying the Directory.

9.3 Each application context is made up of application service elements. These application service elements are defined to use the Remote Operations Service (ROS) of Recommendation X.219 to structure and support their interactions. Thus the DAP and DSP are defined as sets of remote operations and errors using the ROS notation.

ANNEX A

(to Recommendation X.500)

APPLYING THE DIRECTORY

This annex is not an integral part of this Recommendation.

A.1 *The Directory environment*

Note - In this §, the term "network" is used with its general meaning to denote the set of interlinked systems and processes relevant to any telecommunications service, not only one which relates to the OSI network layer.

The Directory exists in and provides services in the following environment:

a) many telecommunications networks will be on a large scale, and will constantly undergo change:

   1) objects of various kinds will enter and leave the network without warning and may do so either singly or in groups;

   2) the connectivity of the objects (particularly network nodes) will change, owing to the addition or removal of paths between them;

   3) various characteristics of the objects, such as their addresses, availability, and physical locations, may change at any time;

b) although the overall rate of changes is high, the useful lifetime of any particular object is not short. An object will typically be involved in communications much more frequently that it will change its address, availability, physical location, etc.;

c) the objects involved in current telecommunications services are typically identified by numbers or other strings of symbols, selected for their ease of allocation or processing but not for ease of use by human beings.

A.2 *Directory service characteristics*

The need for directory capabilities arises from:

a) the desire to isolate (as far as possible) the user of the network from the frequent changes to it. This can be accomplished by placing a "level of indirection" between the users and the objects with which they deal. This involves the users referring to objects by name, rather than by, for example, address. The Directory provides the necessary mapping service;

b) the desire to provide a more "user-friendly" view of the network. For example, the use of aliases, the provision of "yellow-pages" (see A.3.5) etc., helps to relieve the burden of finding and using network information.

The Directory allows users to obtain a variety of information about the network, and provides for the maintenance, distribution and security of that information.

A.3 *Patterns of use of the directory*

Note - This subclause is concerned only with Directory retrieval: it is assumed that the Directory modification services are used solely to maintain the DIB in the form necessary for the application over time.

A.3.1 *Introduction*

The Directory service is defined in these standards in terms of particular requests that a DUA can make and the parameters of them. An application designer is likely, however, to think in more goal-oriented terms when considering the information retrieval requirements of the Directory in that application. Accordingly, this clause describes a number of high-level patterns of use of the Directory service that are likely to be relevant to many applications.

A.3.2 *Look-up*

The straight Directory look-up is likely to be the most frequent type of query of the Directory. It involves the DUA supplying the distinguished name of an object, together with an attribute type. The Directory will return any value(s) corresponding that attribute type. This is a generalization of the classic directory function, which is obtained when the attribute type requested corresponds to a particular type of address. Attribute types for various kinds of address are standardized, including OSI PSAP address, Message Handling O/R address, and telephone and telex numbers.

Look-up is supported by the read service, which also provides the following further generalizations:

- look-up can be based upon names other than the distinguished name of the object, e.g. aliases;
- the values from a number of attribute types can be requested with a single request: the extreme case being that the values of all attributes in the entry are to be returned.

A.3.3 *User-friendly naming*

Names can be given to objects in such a way as to maximize the chances that these names can be predicted (or perhaps remembered) by human users. Names which have this property would typically be made up of attributes which are somehow inherent to the object, rather than being fabricated for the purpose. The name of an object will be common among all of the applications which refer to it.

A.3.4 *Browsing*

In many human-oriented uses of the Directory, it may not be possible for the user (or DUA) to directly quote a name, user-friendly or otherwise, for the object about which information is sought. However, perhaps the user will "know it when he sees it". The browsing capability will allow a human user to wander about the DIB looking for the appropriate entries.

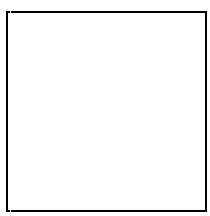
Browsing is accomplished by combinations of the list and search services, possibly in conjunction with read (although the search service includes the capability of read).

A.3.5 *"Yellow Pages"*

There are a variety of ways to provide a "Yellow Pages" type capability. The simplest is based upon filtering, using assertions about particular attributes whose values are the categories (e.g. the "Business Category" attribute type defined in Recommendation X.520). This approach does not require any special information being set-up in the DIT, except to ensure that the requisite attributes are present. However, in the general case, it may be expensive to search where there is a large population because filtering requires the generation of the universal set which is to be filtered.

An alternative approach is possible, based upon the setting up of special subtrees, whose naming structures are designed especially for "Yellow Pages" type searching. Shown in Figure A1/X.500 is an example of a "Yellow Pages" subtree populated by alias entries only. In reality, the entries within the "Yellow Pages" subtrees may be a mixture of object and alias entries, so long as there exists only one object entry for each object stored in the Directory.

FIGURE A-1/X.500 - T0704300-88



A.3.6 *Groups*

A group is a set whose membership can change over time by explicit addition and removal of members. The group is an object, as are its members. The Directory can be requested to:

- indicate whether or not a particular object is a member of a group;
- list the membership of a group.

Groups are supported by having the entry for the group contain a multiple valued "Member" attribute (such an attribute type is defined in Recommendation X.520). The two capabilities mentioned can then be carried out by means of compare and read respectively.

A member of a group could itself be a group, if this is meaningful for the application. However, the necessary recursive verification and expansion services would have to be created by the DUA out of the non-recursive versions provided.

A.3.7 *Authentication*

Many applications require the objects taking part to offer some proof of their identify before they are permitted to carry out some action. The Directory provides support for this authentication process. (As a separate matter, the Directory itself requires its users to authenticate themselves, so as to support access control).

The more straightforward approach to authentication, called "simple authentication", is based upon the Directory holding a "User Password" attribute in the entry for any user that wishes to be able to authenticate itself to a Service. At the request of the Service, the Directory will confirm or deny that a particular value supplied is actually the user's password. This avoids the user needing a different password for every Service. In cases where the exchange of passwords in a local environment that uses simple authentication is considered to be inappropriate, the Directory optionally provides means to protect those passwords against replay or misuse by a one way function.

The more complex approach, called "strong authentication" is based upon public key cryptography, where the Directory acts as a repository of users' public encryption keys, suitably protected against tampering. The steps that users can take to obtain each other's public keys from the Directory, and then to authenticate with each other using them, are described in detail in Recommendation X.509.

A.4 *Generic applications*

A.4.1 *Introduction*

There are a number of generic applications which can be imagined as implicitly supported by the Directory: applications which are not specific to any particular telecommunications service. Two such applications are described herein: the inter-personal communications directory and the inter- system communications directory (for OSI).

Note - Authentication, described in the previous subclause as an "access pattern" could alternatively be thought of as a generic Directory application.

A.4.2 *Inter-personal communications*

The intent of this application is to provide humans or their agents with information on how to communicate with other humans, or groups thereof.

The following classes of objects are certainly involved: person, organizational role and group. Many other classes are involved too, perhaps in a less direct way, including: country, organization, organizational unit.

The attribute types concerned, other than those used in naming, are generally the addressing attributes. Typically the entry for a particular person will have the addresses corresponding to each of the communication methods by which that person can be reached, selected from an open-ended list which includes at least the following: telephony, electronic mail, telex, ISDN, physical delivery (e.g. the postal system), facsimile. In some cases, such as electronic mail, the entry will have some additional information such as the types of information which the user's equipment can handle. If authentication is to be supported, then User Password and/or Credentials will be needed.

The naming schemes used for the various object classes should be user-friendly, with aliases being set up as appropriate to provide alternative names, provide continuity after a name change, etc.

The following access patterns will be manifested in this application: look-up, user-friendly naming, browsing, "Yellow Pages", and groups. To varying degrees, authentication will also be used.

A.4.3 *Inter-system communications (for OSI)*

According to the OSI Reference Model, two Directory functions are required in OSI, one, operating in the application layer, which maps application-titles onto presentation-addresses, and one, in the network layer, which maps NSAP-addresses onto SNPA-addresses (SNPA = Subnetwork Point of Attachment).

Note - For the remainder of this , only the application layer case is dealt with.

This function is carried out by consulting the Directory if the information required to accomplish the mapping is not conveniently available by other means.

The users are application-entities and the object classes of interest are also applicationentities, or subclasses thereof.

The main attribute type concerned, other than those used for naming, is the presentationaddress. Other attribute types, not viewed as necessary for the directory function itself, could support verifying or finding out the application entity type, or the lists of application contexts, abstract syntaxes, etc. supported. The authentication-related attribute types could also be relevant.

The main access pattern to be manifested will be look-up.