INTERNATIONAL STANDARD

ISO/IEC 9594-10

First edition
1998-12-15

Information technology — Open Systems Interconnection — The Directory: Use of systems management for administration of the Directory

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — L'Annuaire: Emploi de la gestion-systèmes pour l'administration de l'Annuaire

Reference number
ISO/IEC 9594-10:1998(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Published by ISO in 2000

Printed in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 9594 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 9594-10 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*, in collaboration with ITU-T. The identical text is published as ITU-T Recommendation X.530.

ISO/IEC 9594 consists of the following parts, under the general title *Information technology — Open Systems Interconnection — The Directory*:

— *Part 1: Overview of concepts, models and services*

— *Part 2: Models*

— *Part 3: Abstract service definition*

— *Part 4: Procedures for distributed operation*

— *Part 5: Protocol specifications*

— *Part 6: Selected attribute types*

— *Part 7: Selected object classes*

— *Part 8: Authentication framework*

— *Part 9: Replication*

— *Part 10: Use of systems management for administration of the Directory*

Annex A forms a normative part of this part of ISO/IEC 9594.

Introduction

This Recommendation | International Standard, together with other Recommendations | International Standards, has been produced to facilitate the interconnection of information processing systems to provide Directory services. A set of 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.

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.

The purpose of Directory management is to assure that needed, accurate Directory information is available to users as scheduled with the expected response time, integrity, security and level of consistency. Furthermore, systems management may be accomplished with the minimum burden on processing time and memory on platforms and the communications system.

The Directory may support open systems applications such as message handling systems, File, Transfer, Access, and Management (FTAM) systems, and transaction processing systems. Therefore, the Directory system may be manageable from an integrated system management platform.

Implementors should note that a defect resolution process exists and that corrections may be applied to this Recommendation | International Standard in the form of technical corrigenda. The identical corrections will be applied to this Recommendation | International Standard in the form of an Implementor's Guide. A list of approved technical corrigenda for this Recommendation | International Standard can be obtained from the Subcommittee secretariat. Published technical corrigenda are available from your national standards organization. The Implementor's Guide may be obtained from the ITU Web site.

Annex A, which is an integral part of this Recommendation | International Standard, defines the managed objects used for Directory System Agent administration.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – THE DIRECTORY: USE OF SYSTEMS MANAGEMENT FOR ADMINISTRATION OF THE DIRECTORY

SECTION 1 – GENERAL

1 Scope

This specification describes the requirements for Directory management, and analyses these requirements to identify those that may be realised by OSI Systems Management services (and protocols), those that are realised by Directory services (and protocols), and those that are realised by local means.

Based on the requirements, this specification defines a model for Directory management that encompasses all of the requirements.

Management of the Directory is divided into four major segments:

a) management of the DIT Domain: Management of Directory information;

b) management of the operation of a single DSA within a DMD;

c) management of the operation of a single DUA within a DMD; and

d) management of the Directory Management Domain (DMD): Integrated management of the functional components of the Directory.

This Recommendation | International Standard covers items a), b) and c). Item d), Management of the Directory Management Domain, is for further study.

Based on the model, this Specification describes the detailed OSI Systems Management Managed Objects used to manage Directory System Agents (DSAs) and Directory User Agents (DUAs) within a Directory Domain, and describes the detailed OSI Systems Management Managed Objects used to manage the interfaces to DUAs and DSAs in other domains as shown in Figure 1.

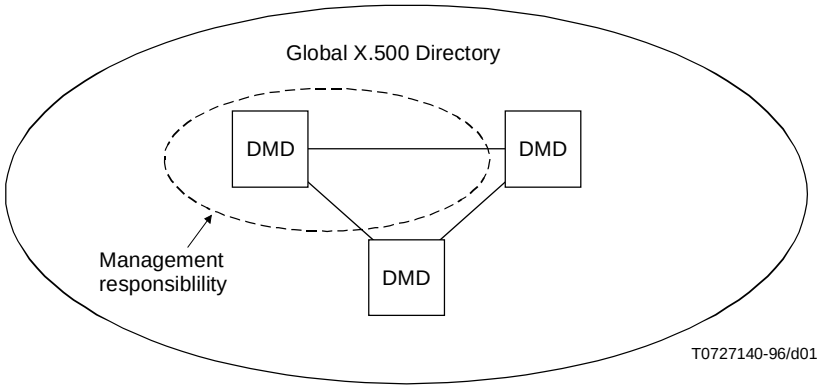

Figure 1 – Scope of Directory management

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardisation Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.*
- ITU-T Recommendation X.500 (1997) | ISO/IEC 9594-1:1997, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services.*
- ITU-T Recommendation X.501 (1997) | ISO/IEC 9594-2:1997, *Information technology – Open Systems Interconnection – The Directory: Models.*
- ITU-T Recommendation X.509 (1997) | ISO/IEC 9594-8:1997, *Information technology – Open Systems Interconnection – The Directory: Authentication framework.*
- ITU-T Recommendation X.511 (1997) | ISO/IEC 9594-3:1997, *Information technology – Open Systems Interconnection – The Directory: Abstract service definition.*
- ITU-T Recommendation X.518 (1997) | ISO/IEC 9594-4:1997, *Information technology – Open Systems Interconnection – The Directory: Procedures for distributed operation.*
- ITU-T Recommendation X.519 (1997) | ISO/IEC 9594-5:1997, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications.*
- ITU-T Recommendation X.520 (1997) | ISO/IEC 9594-6:1997, *Information technology – Open Systems Interconnection – The Directory: Selected attribute types.*
- ITU-T Recommendation X.521 (1997) | ISO/IEC 9594-7:1997, *Information technology – Open Systems Interconnection – The Directory: Selected object classes.*
- ITU-T Recommendation X.525 (1997) | ISO/IEC 9594-9:1997, *Information technology – Open Systems Interconnection – The Directory: Replication.*
- ITU-T Recommendation X.701 (1997) | ISO/IEC 10040:1997, *Information technology – Open Systems Interconnection – Systems management overview*.
- CCITT Recommendation X.720 (1992) | ISO/IEC 10165-1:1993, *Information technology – Open Systems Interconnection – Structure of management information: Management Information Model.*
- CCITT Recommendation X.721 (1992) | ISO/IEC 10165-2:1992, *Information technology – Open Systems Interconnection – Structure of management information: Definition of management information*.
- CCITT Recommendation X.722 (1992) | ISO/IEC 10165-4:1992, *Information technology – Open Systems Interconnection – Structure of management information: Guidelines for the definition of managed objects.*
- ITU-T Recommendation X.723 (1993) | ISO/IEC 10165-5:1994, *Information technology – Open Systems Interconnection – Structure of management information: Generic management information.*

2.2 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation X.700 (1992), *Management framework for Open Systems Interconnection (OSI) for CCITT applications.*

  ISO/IEC 7498-4:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management framework.*

- CCITT Recommendation X.710 (1991), *Common management information service definition for CCITT applications.*

  ISO/IEC 9595:1991, *Information technology – Open Systems Interconnection – Common management information service definition.*

– CCITT Recommendation X.711 (1991), *Common management information protocol specification for CCITT applications.*

ISO/IEC 9596-1:1991, *Information technology – Open Systems Interconnection – Common management information protocol – Part 1: Specification.*

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 OSI Reference Model definitions

The following terms are defined in ITU-T Rec. X.200 | ISO/IEC 7498-1:

a) application-entity;

b) Application Layer;

c) application-process;

d) application protocol data unit.

3.2 Management Framework definitions

The following terms are defined in CCITT Rec. X.700 and ISO/IEC 7498-4:

a) management information base;

b) managed object.

3.3 System Management Overview definitions

The following terms are defined in ITU-T Rec. X.701 | ISO/IEC 10040:

a) agent;

b) manager;

c) notification;

d) managed object class.

3.4 Management Information Model definitions

The following terms are defined in CCITT Rec. X.720 | ISO/IEC 10165-1:

a) behaviour;

b) conditional package;

c) inheritance;

d) naming tree;

e) package;

f) subclass;

g) superclass.

3.5 Directory Model definitions

The following terms are defined in ITU-T Rec. X.501 | ISO/IEC 9594-2:

a) access control;

b) Administration Directory Management Domain;

c) alias;

d) attribute;

e) attribute type;

f) attribute value;
g) authentication;
h) Directory Information Tree ;
i) Directory Management Domain ;
j) Directory System Agent ;
k) DSA Specific Entry;
l) Directory User Agent (DUA);
m) distinguished name;
n) entry;
o) name;
p) object (of interest);
q) Private Directory Management Domain;
r) relative distinguished name;
s) root;
t) schema;
u) security policy;
v) subordinate object;
w) superior entry;
x) superior object;
y) tree;
z) (Directory) user.

3.6 Distributed Operation definitions

The following terms are defined in ITU-T Rec. X.518 | ISO/IEC 9594-4:

a) hierarchical operational binding;
b) non-specific hierarchical operational binding.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

ADDMD	Administration Directory Management Domain
CMIP	Common Management Information Protocol
DAP	Directory Access Protocol
DIB	Directory Information Base
DISP	Directory Information Shadowing Protocol
DIT	Directory Information Tree
DMD	Directory Management Domain
DOP	Directory Operational Binding Management Protocol
DSA	Directory System Agent
DSE	DSA Specific Entry
DSP	Directory System Protocol
DUA	Directory User Agent
HOB	Hierarchical Operational Binding
MIB	Management Information Base

NHOB	Non-specific Hierarchical Operational Binding
NSAP	Network Service Access Point
NSSR	Non-specific Subordinate Reference
OSI	Open Systems Interconnection
PRDMD	Private Directory Management Domain
RDN	Relative Distinguished Name
TMN	Telecommunications Management Network

5 Conventions

With minor exceptions, this Directory Specification has been prepared according to the "Presentation of ITU-T | ISO/IEC common text" guidelines in the Guide for ITU-T and ISO/IEC JTC 1 Co-operation, March 1993.

The term "Directory Specification" (as in "this Directory Specification") shall be taken to mean this Recommendation | International Standard. The term "Directory Specifications" shall be taken to mean all of the X.500-Series Recommendations | ISO/IEC 9594.

This Directory Specification uses the term "1988 edition systems" to refer to systems conforming to the first (1988) edition of the Directory Specifications, i.e. the 1988 edition of the series of CCITT X.500 Recommendations and the ISO/IEC 9594:1990 edition. This Directory Specification uses the term "1993 edition systems" to refer to systems conforming to the second (1993) edition of the Directory Specifications, i.e. the 1993 edition of the series of ITU-T X.500 Recommendations and the ISO/IEC 9594:1995 edition. Systems conforming to this third edition of the Directory Specifications are referred to as "1997 edition systems".

This Directory Specification presents ASN.1 notation and Managed Object Definitions in the bold Times Roman, 9 point typeface. When ASN.1 types and values or Managed Object Definitions are referenced in normal text, they are differentiated from normal text by presenting them in the bold Times Roman, 9 point typeface. Access control permissions are presented in italicized Helvetica.

SECTION 2 – MANAGEMENT REQUIREMENTS

6 Directory management requirements

The collection and processing of management information is an overhead set against the primary objective of the Directory. Consequently, it is essential to ensure that all activities involved in acquiring management information are useful, valid and present the minimum overhead to the natural processes of Directory components.

In order to derive the required management information and associated actions, it is necessary to analyse the various entities which both provide the Directory service and also interact with it so that the relevant management needs are identified. Furthermore, the Directory will operate in conjunction with other networks and services. The Telecommunications Management Network[1)] (TMN) is designed to provide a framework for management across differing networks and services. Hence, the management features of Directory components are aligned with the expectations of TMN.

6.1 Introduction

This Section analyses the environment in which a Directory will operate and isolates the management requirements.

The management requirements are defined by analysis of the activities of roles concerned with using, operating and owning a Directory service. The motivation for the selection of these roles has been influenced by the functional hierarchy view of management, defined within the TMN. This takes a broad view of an organisation offering Directory services and encompasses the need for low-level component management, the customer-oriented requirements of offering services and the effects of the business objectives of the owners of Directory systems.

1) ITU-T Rec. M.3010, Principles for a telecommunications management network.

6.2 Sources of management requirements

6.2.1 Service agreement

6.2.1.1 Directory customer service agreement

A Directory service agreement is a set of terms and conditions governing the provision of the Directory services and establishing the contractual relationship between the Directory customer and a Directory service provider. A service agreement may cover a number of items relating to the expected operation of the Directory, such as accessible Directory information (including maintenance of indirect data links such as **seeAlso** attributes and **groupOfName** entries), allowed operations on accessible Directory information, quality of service operation, conditions for settlement for usage of the service, and availability of the service and access points.

Of these items, some are directly embodied by Directory components and management activities (for example, detecting aliases that point to non-existent distinguished names). Conversely, some service agreement items (for example, settlement) are indirectly embodied by Directory components in that a management process uses a record of Directory component activity as a basis for fulfilling the service agreement.

Associated with a service agreement there are a number of roles such as:

- Directory user;
- Directory customer;
- Directory service manager;
- Directory system manager/administrator (see 6.2.2); and
- Directory business manager (see 6.2.3 and 6.3.5).

A Directory customer, acting on behalf of Directory users, enters into an agreement with the Directory management organisation which determines an agreed service to be presented to users. The Directory customer may represent any arbitrary group of users, the structure and content of which are not restricted by the Directory management organisation.

A Directory user is a consumer of Directory services. Actions of Directory users stimulate the Directory components to produce management information in order that the Directory service manager may ascertain whether the Directory is operating within the bounds of the Directory user's service agreement.

A Directory service manager is responsible for ensuring that a service agreement is implemented and maintained. The Directory service manager functions may encompass a number of areas such as:

- registration (e.g. of Directory users, Directory customers);
- configuration changes (e.g. enabling or disabling DSAs);
- assistance (e.g. help desk, technical support);
- service configuration changes (e.g. changes to service characteristics);
- quality of service monitoring and reporting; and
- accounting, billing and settlement.

6.2.1.2 Peer service provider service agreement

In order to fully satisfy a service offered to a Directory user, it may be necessary to make use of Directory services provided by other Directory service providers. The essence of a peer service provider service agreement may be similar to that constructed for the basis of interaction with Directory users. That is, available information, allowed operations, access details, etc., will need to be agreed between two Directory service providers before interaction can occur.

6.2.2 Operations

An essential part of attaining an agreed service is the on-going monitoring and maintenance of the Directory components which provide the service:

- Re-configuration of Directory components:
  a) predictable downtime due to equipment maintenance;
  b) unpredictable downtime due to equipment failure.

- Management re-configuration:
  - for example, redirecting collected management information out of office hours.
- Management of product operating limitations:
  a) observing product maximum operating parameters (e.g. maximum number of associations for a DSA, maximum number of entries for a DSA);
  b) observing inter-provider operating parameters.
- Troubleshooting:
  - configuring components to act in a specific way for the purposes of problem solving.

The role associated with Operations is Directory system administrator.

6.2.3 Business processes

Business processes reflect the activities undertaken by business managers in the pursuit of business objectives through the offering of Directory services. Objectives and motivations differ from one organisation to another, e.g. financial gain is one motivation. Different objectives/motivations will result in different sets of management information being relevant to different organisations. The Directory management facilities shall enable the construction of management policies by organisations.

Information regarding the performance against those objectives is required. Activities which are undertaken will include sell/(advertise) services, expand/contract system, procure equipment, and evolve services.

The role associated with business processes is the Directory business manager who will strive to meet business objectives through the setting of service targets (for example, in terms of reducing operating costs), selling/advertising services, expanding/contracting capacity, procuring capital equipment, instigating new service offerings, etc.

6.3 Analysis of management requirements

The identification of management requirements illustrates the roles and activities concerned with both using, providing and owning a Directory service. A closer analysis of these roles and activities will identify a set of required management information and management actions which serve to maintain a successful Directory service.

6.3.1 General requirements

There are a number of issues to consider:

- Management information can be expressed in a number of different forms such as maintaining logs and counters, establishing gauges and thresholds, and generating events and alarms. It is expected that the management system will supply standardised mechanisms for the expression of different management information formats.
- Management activities, and thus the need for specific elements of management information, may vary over time. There is a need for the dynamic configuration of the collection of management information.
- Implementation of management policies should not be hindered by Directory management specifications.
- Operational information produced by Directory systems may change status according to which type of organisation is operating the service and which type of service agreement has been made.

6.3.2 Directory user

6.3.2.1 Allowed Directory user activity

6.3.2.1.1 Successful Directory user access

Record Directory DAP, DSP, DISP, DOP activity:

- Log operation counts.
- Log operation details.
- Log details against the data retrieved rather than the operation invoked.
- Log resource usage.
- Notifications of an exceptional valid operation that will take place may be required. This may be required if, for example, the operation would cause a large amount of activity within the Directory system (e.g. a subtree search at the country level, or a shadow update is occurring).

6.3.2.1.2 Unsuccessful Directory user access

Directory reports no errors, but service operation is not as expected. It will be necessary to report the details to service management as a violation of the service agreement. The Directory components will only collect management information as described in 6.3.2.1.1.

The unexpected event may be against any of the items of the service agreement that the user is aware of, for example:

- unable to invoke a specific Directory operation on the DIB;
- returned data is not of a quality agreed within the service agreement (e.g. the data is out of date or certain agreed optional attributes are not included).

A condition caused by a valid operation which fails because of:

- direct information failure (e.g. alias dereference failure, knowledge problem);
- indirect information failure (e.g. an entry does not exist with the distinguished name found on a previous read of a **groupOfNames** entry or **seeAlso**);
- equipment failure.

6.3.2.2 Disallowed Directory user activity

6.3.2.2.1 Disallowed unsuccessful Directory service access

The Directory detects and shall notify an attempt at illegal access to:

- the Directory service (i.e. the bind);
- specific information and (invocation of) operations (i.e. detection by access control procedures).

Logging of all unauthorised activity may also occur.

Additionally, resource usage incurred when making an unauthorised access can be. This information allows system and service administrators to assess the cost of unauthorised access.

6.3.2.2.2 Disallowed successful Directory service access

This situation occurs when a Directory user has successfully accessed the Directory in a way which breaches the service agreement but the Directory did not detect this as an error. This indicates an error in the system configuration against the service agreement. Detection would only take place if sufficient log information was available and was analysed off-line.

6.3.3 Directory customer

- Establishment of service agreement:
  - scope assigned to the user (i.e. anywhere, within the DMD, within the DSA).
- Represents users of service – The specific combination of users in terms of numbers, structure and service agreement features is arbitrary and not inhibited by the Directory management capabilities.
- Query status of service against service agreement.
- Query capabilities of service with a view to extending/curtailing existing service agreement.
- Settle for usage of service. Settlement arrangement is based upon an internal calculation of service management and can include:
  a) query-oriented, based on resources used in querying;
  b) data supplier oriented, based on resources used by information residing in DIT;
  c) pre-defined absolute time limit usage of the Directory (as opposed to a specific association time);
  d) a pre-settled resource usage of the Directory.

The customer may represent a number of users; the settlement process will need to be able to identify users with the billable customer.

6.3.4 Directory service manager

The Directory service manager acts upon requests made by Directory customers and the need to monitor the operation of the Directory service in order that service agreements are maintained:

- Create Directory configuration necessary for meeting a service agreement.
- Respond to requests for service information from customer:
  a) Billing information – Based upon customer, rather than user.
  b) Problem reports.
- Make decision as to exactly what management information is required to be collected and when, in order that the service agreement is maintained.
- Inhibit binds (for example, due to user not registered for a service, service available during limited times, service unavailable due to customer/service contravening service agreement).
- Validate operation requests against service agreements.

When considering the management of Directory information extraction, there are a number of issues:

- Control is needed over the amount of data that may be extracted, and the Directory currently addresses this concern through the setting of size and time limits on requests. Additionally, control may be imposed on users who would otherwise attempt to destroy the integrity of information within the Directory.
- Waste of resources through either retrieval of Directory information based upon an inappropriate choice of filter, which results in a large number of entries being processed (e.g. search using a substring filter of "Hotel" within a DIT subtree holding UK data).
- Waste of resources through specification of an operation that it is known will not succeed (e.g. searching for an entry with a **localityName** filter which is non-existent).
- Attempting to retrieve directory information on an illegal basis. This may either be through the usage of a particular attribute type within a filter (e.g. filtering entries against their telephone number is not allowed within the UK) or through the use of a particular matching algorithm (e.g. it is not permitted within France to use final substring filter match on surnames).
- A Directory service provider may not allow a user a wide-ranging browsing capability. This would result in a designated set of DIT access locations (distinguished names).

6.3.5 Directory business manager

- Specify monitoring conditions concerned with detecting:
  a) specific usage patterns (for example, from particular known groups or geographical areas);
  b) candidates for the expansion/contraction of service resources (either through procurement or reconfiguration) – due to demand;
  c) candidates for identifying groups of users which do/don't use particular service features (as basis of marketing exercises);
  d) configuration to cope with localised (temporally and/or geographically) demand (e.g. for special events).

6.3.6 Management of the DIT domain

The schema, including DIT, object classes, attributes, attribute syntaxes, structure rules and matching rules shall be implemented and maintained. The schema may be "published" in subentries. Provision is made for adding, modifying and deleting Directory names and entry information for both user and operational attributes. The Directory administrator ensures that relative distinguished names are registered and that the contents of entries are correct and in accordance with the schema. Tools for content error detection and analysis should be available.

6.3.6.1 Management of aliases and other pointers

Management of aliases and similar pointers is not standardised. The Directory system manager may require solutions to ensure consistency between object entries and alias entries. That is, it should be possible for a manager to list those aliases the target entries of which do not exist.

6.3.6.2 Management of lists and seeAlso

Consistency between Directory lists and **seeAlso** attributes may be managed; that is, there should be an entry for each member of a list and the entry named in a **seeAlso** attribute should exist. The Directory Specifications do not provide this service. An examples of a list is **groupOfNames**, defined in these Directory Specifications.

6.3.7 Management of a DSA

Requirements for management of the DSA application process may be divided into the accounting, configuration, fault, performance and security functional areas; the same information may be applicable to more than one functional area. The management requirements can be divided further into those that can be considered monitoring and those that can be considered controlling. Some notifications from the managed system may require real-time reporting to a manager and others may be logged for future analysis.

6.3.7.1 Configuration management

Configuration management is the maintenance and exchange of information with regards to actual physical and logical placement of the components of a system. With regards to Directory management, requirements for the management of the Directory information should be distinguished from requirements for the management of the DSA:

- Requirements for the management of the Directory information: some of the important requirements are:
  - a) provide the capability to ensure that the Directory information is configured according to the appropriate subschema; and
  - b) provide the capability to manage the subschemas including adding, deleting and modifying the subschema;
  - c) provide tools to redistribute the Directory Information Base to other DSAs.
- Requirements for the management of the DSA: some of the important requirements are:
  - a) provide the capability to initiate user service, e.g. registering a user with a DUA and setting some of the default service control parameters;
  - b) provide the capability of managing inventory and location of deployed Directory components (inventory to be managed includes software resource details, license details, and vendor contact information);
  - c) provide the Directory managers with the capability to configure, add, or delete components, as well as the capability to enable (e.g. starting a DSA process) or disable Directory entities;
  - d) provide the Directory managers with the capability to lock and unlock the Directory;
  - e) provide the capability to list the operational bindings of which the DSA is cognisant, and to which other DSAs can make an application association or to which a referral can be returned;
  - f) provide the ability to reconfigure the DSA to improve performance and/or overcome faults;
  - g) accommodate topology changes;
  - h) provide the ability to examine and be notified of changes of state, monitor overall operability, and usage of the DSA;
  - i) provide the controls for the monitoring and distribution of configuration information to other DMDs;
  - j) provide information for neighbour DSAs including: DSA name and security credentials, presentation address, lower layers supported, naming contexts and availability;
  - k) provide the ability to summarise shadowing agreements;
  - l) provide the ability to set administrative limits and thresholds (e.g. maximum time for an operation, maximum number of associations); and
  - m) provide the capability to configure support for matching rules and attribute syntaxes in the DSA.

6.3.7.2 Fault management

Fault management deals with identifying, isolating, reporting, and correcting faults arising in a system. With regards to Directory management, requirements for the management of the Directory information should be distinguished from requirements for the management of the DSA:

- Requirements for the management of the Directory information: some of the important requirements are:
  - a) provide the capability to report errors (such as **nameError**, **attributeError** or **updateError**) returned from the invocation of a Directory operation to a Directory manager (note the DAP only returns errors to a Directory user);

b) provide the capability for a Directory user to report any inconsistency in the returned Directory information (such as missing mandatory attributes or improper attribute values) to a Directory manager;

c) provide the capability to log and analyse errors mentioned above; and

d) provide for remote back-up of Directory information.

– Requirements for the management of the DSA: some of the important requirements are:

a) provide the capability to detect and report failures in the Directory service, including connectivity failures and failures of any Directory operations, or any of the Directory system components;

b) provide the capability to recover from faults (e.g. via reconfiguration or back-up of selected components);

c) provide the capability to log and analyse faults (e.g. fault correlation).

d) provide the capability to interact with other management areas such as configuration management and performance management;

e) specify fault alarm threshold values and faults to be alarmed;

f) specify types of notifications required;

g) determine frequency of polling for abnormalities;

h) anticipate faults by analysing logged operations;

i) provide the capability to manipulate the stored information in the Directory database such as back-up, restore, audit, resource management, etc.;

j) analyse logged data for operations which have exceeded quality of service for response time; and

k) log knowledge inconsistencies reported during chaining.

6.3.7.3 Performance management

Performance management enables the evaluation of the behaviour of system resources. It provides the functions to gather and disseminate statistical data, maintain historical logs of system performance, and simulate various system modes of operations. With regards to Directory management, requirements for the management of the Directory information should be distinguished from requirements for the management of the DSA:

– Requirements for the management of the Directory information: some of the important requirements are:

a) provide the capability to collect performance data on the usage of Directory information, such as maintaining counters which measure the number of invocations of a Directory operation on an entry;

b) provide the capability to detect severe performance problems, such as placing thresholds on relevant counters; and

c) provide the capability to replicate frequently accessed Directory information to selected DSAs.

– Requirements for the management of the DSA: some of the important requirements are:

a) provide the capability to collect system performance data, such as maintaining counters for the number of Directory operations served by a DSA, and the number of chainings performed by a DSA;

b) provide the capability to detect severe performance problems, such as placing thresholds on relevant counters;

c) provide sufficient knowledge references (such as cross references) for DSAs which have high traffic;

d) provide replication of Directory information to appropriate DSAs;

e) provide performance measurement tools (such as simulation packages) which can be used to measure and optimise system performance;

f) analyse volume and source of requests satisfied locally and those satisfied outside the DSA to satisfy cost and response time;

g) provide tools to analyse trace information;

h) log shadow updates including shadow consumer name;

i) provide statistics to support analysis per entry or per operation; and

j) provide the ability to collect and analyse statistics on operations and associations with neighbour DSAs.

6.3.7.4 Security management

Security management provides the functions to support security services, maintain security logs, and distribute relevant information to other systems. With regards to Directory management, requirements for the management of the Directory information should be distinguished from requirements for the management of the Directory system:

- Requirements for the management of the Directory information: some of the important requirements are:
  a) provide an access control policy to protect Directory information from illegal usage by Directory users; and
  b) provide mechanisms to monitor security threats to Directory information.
- Requirements for the management of the DSA: some of the important requirements are:
  a) provide an access control policy to protect DSA system files from illegal usage by managers/agents;
  b) provide mechanisms to monitor security threats to DSAs;
  c) reporting security violations;
  d) provide an audit trail; and
  e) provide for the establishment and maintenance of a DSA's credentials.

An audit trail of accesses to the Directory should be created and maintained. This audit trail should be protected from unauthorised access, modification and destruction.

The audit system will record security-related events in a manner that will allow detection and/or after-the-fact investigations to trace security violations to the responsible party. A security-relevant event is defined as any event that attempts to change the security state of the system (e.g. change access control information, change user password, etc.) and any event that attempts to violate the security policy of the system, e.g. too many attempts to bind, an attempt to access unauthorised objects, etc.

The audit capabilities shall be configurable by the security officer so that the events audited and the information captured for an event can be enabled or disabled. There shall be provisions for operating although the audit trail has filled to its capacity. An alarm shall notify the security officer that the audit trail is filled. Experience has shown that a great deal of audit information can be generated if one is not selective in the criteria for auditing. One wants flexibility so that, under normal circumstances, little audit information is collected, but that, in the event that suspicion is aroused, more detailed auditing can be enabled.

Each recorded event will include at least the following:

- event time;
- origin of request or response;
- operation;
- target of operation;
- outcome of request or response.

The following security problems are specified in the Directory specifications for operations including binding, and should be logged as security violations for possible inclusion in an audit:

inappropriateAuthentication
invalidCredentials
insufficientAccessRights
invalidSignature
protectionRequired
blockedCredentials
noInformation

6.3.7.5 Accounting management

Accounting management may provide the following functions:

- accounting functions to provide facilities to generate, collect, store and process the customer's accounting information (e.g. the time of usage of services/resources);
- charging and billing functions for the computation/establishment of every individual customer-based tariff and usage records; and
- cost accounting functions for keeping track of costs (for providing the services) and revenues as part of the business management.

SECTION 3 – MANAGEMENT MODELS

7 Directory Management Model

This clause specifies the Directory Management Model.

7.1 Introduction

A Directory Management Model serves three purposes:

a) it identifies the managed objects;

b) it identifies the entities (in terms of their roles and their functionality) involved in Directory management; and

c) it identifies the data flow between the communicating entities.

The objects to be managed depend on the management requirements. For example, a Directory service provider may need to manage objects such as service access points and Directory customers.

7.2 Directory Management Model components

The entities involved in a Directory Management Model include DUAs, DSAs, Directory managers, Directory agents, Directory users, and possibly Directory customers. Some explanations of the entities follow:

– A *Directory user* is a user of the Directory service.

– A *Directory customer* is an organisation or a person representing an organisation, that procures Directory services, which for some reason makes a value judgement on whether the Directory service is doing its function or not. In an organisation, there is typically only one Directory customer but possibly hundreds or thousands of Directory users.

– A *Directory manager* is an entity involved in Directory management. It may play a number of roles. The management roles reflect the management requirements for the DMD. Each role can be mapped to a set of management functions and managed objects. Examples of roles are given below:

  - *Helpdesk*: Will receive requests for help from the Directory users/customers, and attempt to either solve the problem or make another Directory manager solve it.
  - *Subscription manager*: Will process the requests for Directory users to be added or deleted from the Directory system, or to change credentials.
  - *Configuration manager*: Will take care of adding, removing or changing the Directory components.
  - *Fault manager*: Takes actions or passes the problems on by way of the event reports indicating faults in the functioning of the Directory service.
  - *Security manager*: Will protect from the illegal access to the Directory system by intruders.
  - *Accounting manager*: Will decide the reasonable monetary settlement based on the accounting data.
  - *Schema manager*: Will manage the Directory subschemas which includes the DIT structure rules, DIT content rules, name forms, object classes, attribute types, and matching rules.
  - *Planning manager*: Takes decisions, based on statistical data and inputs about the needed changes to be made to the Directory system to meet future demands.

– A *Directory agent* is an entity working on behalf of a Directory manager. It controls one or more Directory managed objects. Quite often, a Directory agent is collocated with a managed Directory entity, e.g. a DSA is collocated with a Directory agent. The relation between Directory managers and Directory agents can be many-to-many. That is, more than one Directory agent can act on behalf of a Directory manager, and a Directory agent can act on behalf of more than one Directory manager. The Directory Management Model should not constrain the Directory management protocols used between a Directory manager and a Directory agent.

7.3 Layered Directory Management Model

It is useful to structure the management functions or roles into layers. An example of a layered model can be derived from the TMN Model. Figure 2 defines such a layered model for Directory management.

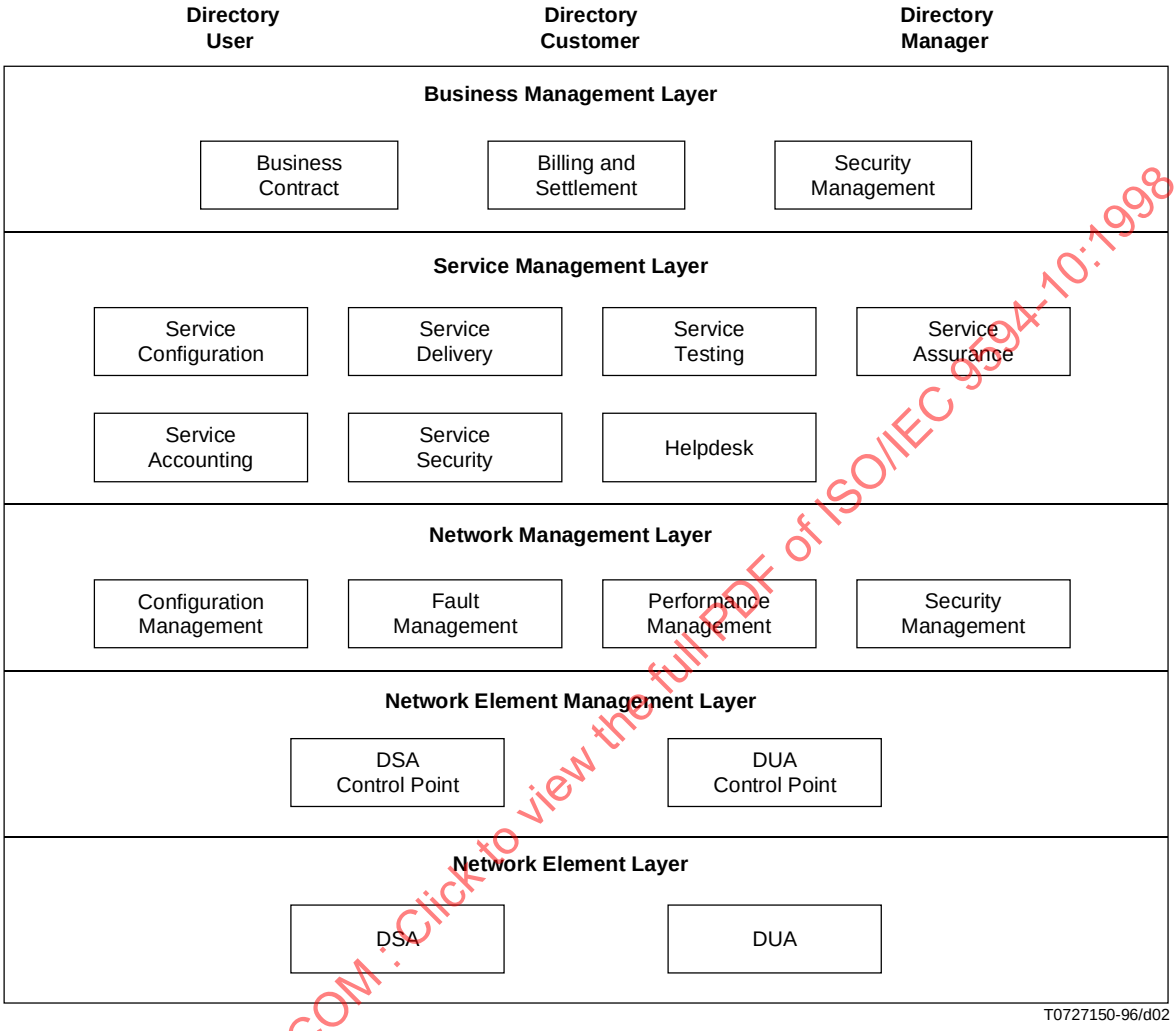

Figure 2 – Layered Directory Management Model

The different layers of this model are defined as follows:

– Layer 1, the *Network Element Layer*, consists of the managed objects chosen for Directory management in a DMD. The Directory Management Information Model (see clause 9) gives a refinement of this layer.

– Layer 2, the *Network Element Management Layer*, shows the different object managers. Logically, there is an object manager for every managed object class. For example, there is a manager for DSAs, a manager for Directory customers, a manager for entries, etc. Note that more than one managed object can be managed by the same manager.

– Layer 3, the *Network Management Layer*, consists of functional area managers involved in the management of the internal resources of a DMD. For example, there is a configuration manager, a fault manager, a security manager, etc. A functional area manager may need the support of more than one object manager in layer 2. For example, a configuration manager needs the support of a DSA manager as well as a DUA manager. Note that accounting managers, whose roles are to manage external resources such as Directory customers, are not included in this layer.

– Layer 4, the *Service Management Layer*, is concerned with the interface to the customer. It includes Accounting Management, the Helpdesk, and Service Contract Management as well as Security Management. The Helpdesk is the human point of contact for assistance with using the service, including the reporting of faults. Service Contract Management is the interface to the customer. It handles the order and sends information to the elements, information necessary to set up an account, for example.

– Layer 5, the *Business Management Layer*, is concerned with a total enterprise (i.e. all service and networks) and carries out an overall business coordination. Between service providers, for example, it would include billing and settlement.

7.4 Directory Information Model and System Management Information Model

Not all the information needed by a Directory manager can be placed in the DIB as operational attributes. Management information which does not change frequently can be safely placed in the DIB. For example, it is unlikely that operational attributes such as **dITStructureRules** and **prescriptiveACI** would change often. The advantage of placing these static operational attributes in the DIB is that a Directory manager can use the Directory Access Protocol to manage the static aspects.

Management information such as counters to monitor traffic for an entry are quite dynamic in nature. The DIB is not designed to hold dynamic information. Thus, dynamic management information should be placed in a Management Information Base. For this reason, every entry in the DIB shall be able to be managed. However, it is not necessarily the case that there is a managed object instance for every entry. The Directory managed entry in Figure 3 shows the related DSE managed object. Note that the name of the Directory entry and the name of the corresponding managed object are quite different from each other. For example, the Relative Distinguished Name of the managed object DSE may be the Directory name of the DSE.

NOTE – Managed object definitions defined in this Specification do not define such dynamic information. However, implementations may choose to extend these management object definitions to hold such information as desired.

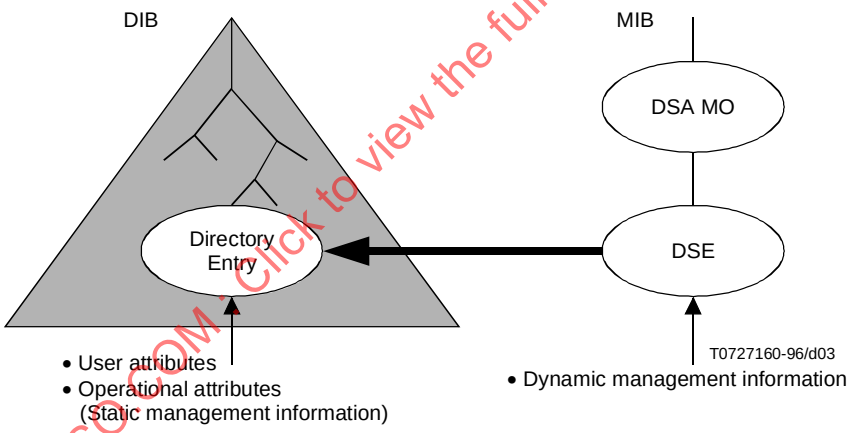

Figure 3 – Directory managed entry

7.5 Directory Service Model

The role that X.500 components play in providing a Directory service to a Directory user has two main aspects. The first aspect is that of maintaining control over the information processing capability of the Directory and the second aspect is that of maintaining control over other service parameters (such as maximum numbers of entries returnable).

The Directory service is a managed object, maintained by the directory service managers, which can be accessed by both a DSA and a DUA. The Directory service MO will specify the information aspects of the service and the control aspects of the service.

7.5.1 Directory Information Service

The Directory information processing capability is defined to be a combination of Directory request parameters, namely:

– Directory Operation;

– attribute type;

- attribute values to be used in filtering; and
- matching rules applied.

In general, the DSA will service a request on any combination of these parameters, whereas it is sometimes (as outlined within Section 2 above) not desirable for the full set of combinations of parameters to be applied to the schema/structure presented by the Directory to the user. Specific combinations of parameters may contravene some policy adhered to by the service management which operates the Directory system. It is, therefore, desirable to be able to instruct X.500 components to prevent requests with these combinations from being executed.

The Directory information processing capabilities of the Directory service has the following components:

- a set of allowed parameter combinations;
- a set of disallowed parameter combinations;
- a set of allowed distinguished names which denote which parts of the DIT are allowed to be specified with a request as either the target object or base object;
- a set of disallowed distinguished names which denote parts of the DIT which cannot be specified within a request as either the target object or base object;
- an identifier for the Directory service; this recognises that a particular Directory system may be used to support a number of different Directory services; and
- a description of the Directory service.

Apart from the service identifier, all other components are not required. Thus, a particular Directory service may contain any or none of the components, depending on how the Directory service manager would like to influence the DSA behaviour.

The allowed and disallowed parameter combinations are expressed as specific combinations of the Directory request parameters listed above. It is expected that a typical Directory service will have a number of these combinations defined.

The set of allowed and disallowed distinguished names which may be supplied limits 'entry points' into the DIT. This is useful for inhibiting Directory requests which result in (for example) full country searching.

7.5.2 Directory Control Service

Directory Control Service covers the parts of the Directory system activity which manages Directory service activity. This includes such activities as:

- limiting the numbers of entries to be returned; and
- limiting the time allowed for return of results.

This information is largely defined as part of the DSA managed object attributes.

8 Provision of management services

Management requirements can be met using a combination of Directory services, the Common Management Information Service, and by local means.

The user attributes in the Directory are usually maintained using DAP. Operational attributes can be maintained using either DAP or the **manageDSAIT** extension to DAP.

DOP is usually used to update knowledge references. Subordinate references, non-specific subordinate references, and immediate superior references, as well as the context prefix information for naming contexts, can be created and maintained by relevant hierarchical operational bindings.

Shadowing is used to create and maintain references in two ways: first, when shadowing agreements are established or terminated, access points are added or removed from the **consumerKnowledge** and optionally the **secondaryShadow** operational attributes. This information can then be used by the relevant hierarchical operational bindings to update the subordinate reference in the superior master DSA and the immediate superior reference in the subordinate master DSA. Second, the DISP propagates the knowledge references held by supplier DSAs to shadow consumer DSAs.

Cross reference distribution is a feature of the DSP; cross references may be returned in chaining results and referrals.

Management protocols may be used to manage the Directory components. For example, the operational status of the DSA may be controlled using the Common Management Information Protocol (CMIP) and the Directory service controls may be managed using CMIP. The management protocols also provide for notifications of events that may be logged or sent to the manager. The logs can be analysed by an application to provide performance and accounting information.

In Figure 4, an example of the use of Directory and management protocols is depicted. The figure shows which protocol is used between components. Where no protocol is shown, such as between a DSA and a CMIS agent, the interface is not standardised.

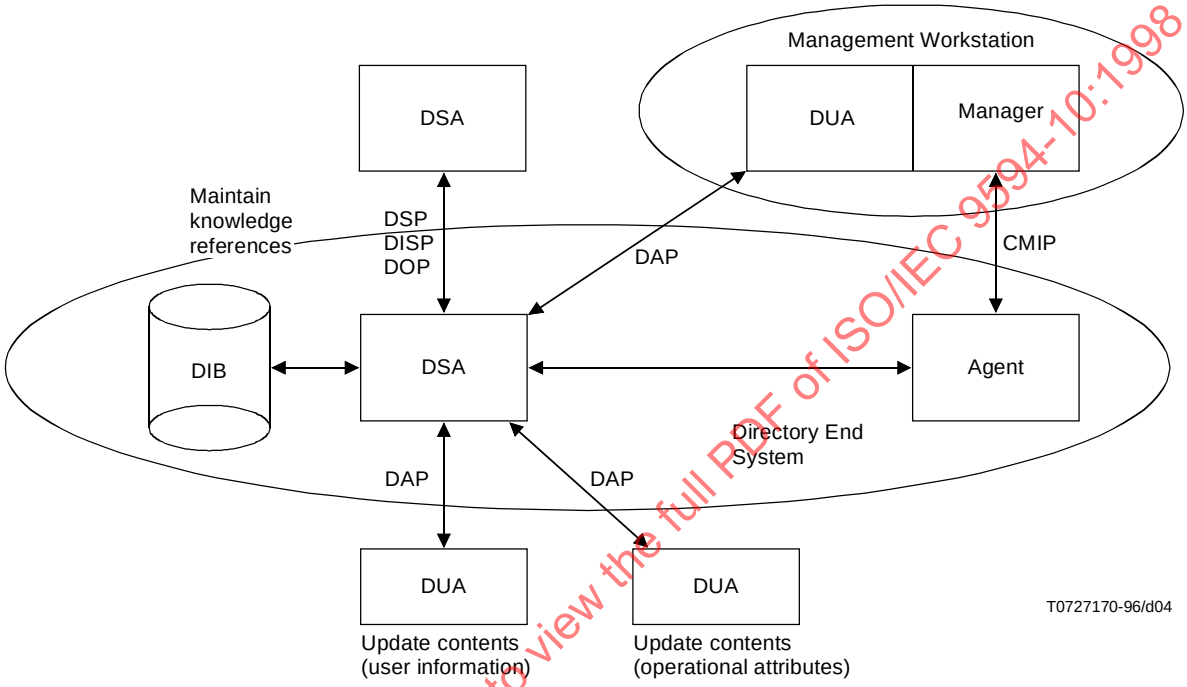

Figure 4 – Entities in Directory Management

A management workstation, which includes both a Common Management Information Service (CMIS) manager and a Directory User Agent, presents a single-user interface to the DSA administrator. Such a management workstation uses the appropriate service and protocol to affect the management operations requested by the DSA administrator.

A DSA administrator may also use a DUA to set values of DSA-specific or DSA-shared attributes in the DSA Information Tree.

A DSA administrator may also use the CMIS manager to start and stop the DSA or to initiate a shadowing operation.

Figure 4 also shows two other DUAs. One is used by a user or Directory administrator to maintain user attributes, and the other is used by a Directory administrator to maintain operational attributes, such as Access Control Information (ACI).

Shadowed information is maintained using DISP. The DSP, DISP and DSP may be used to update knowledge references as explained above.

The use of Directory services to satisfy some of the requirements which are identified in clause 6 is described elsewhere in these Directory specifications. The use of management protocols to satisfy some of the requirements involves a Management Information Model as described in clauses 9 and 10. The use of local means to satisfy some of the requirements is outside the scope of these Directory specifications.

9 Directory Management Information Model

The Directory Management Information Model is a refinement of Layer 1, i.e. Network Element Management Layer, of the layered Directory Management Model. It defines a structure for the Directory managed object classes that are used in management protocols to address some of the requirements which are identified in clause 6. It also gives a formal definition of each Directory managed object class. The managed objects for Directory management are described in clause 10, and GDMO definitions are specified in Annex A.

Figure 5 shows the Directory Management containment hierarchy for a managed Directory system. Conceptually, underneath the DMD, we find DSAs and Directory customers as components. Underneath DSAs, we find known DSAs, known DUAs, DSEs, NHOBs, HOBs and Shadowing Agreements as further subcomponents:

- *DMD* represents a Directory Management Domain;
- *DSA* represents a DSA within the DMD;
- *Directory Customer* represents a Directory customer of the DMD;
- *Directory Service* represents a managed object to manage the Directory service provided to a Directory Customer;
- *Directory User* represents a managed object to manage a single user of a Directory Customer;
- *DSE* represents a managed object to manage a DSE, including knowledge, subentries and entries;
- *Known DSA* represents the containing managed object's view of a peer DSA;
- *Known DUA* represents the containing managed object's view of a peer DUA;
- *UL Connection End* represents an application association between a DSA and a DSA/DUA;
- *HOB* represents a managed object to manage hierarchical operational binding;
- *NHOB* represents a managed object to manage non-specific hierarchical operational binding;
- *Shadowing Agreement* represents a managed object to manage shadowing agreement;
- *DUA* represent a managed object to manage a DUA.

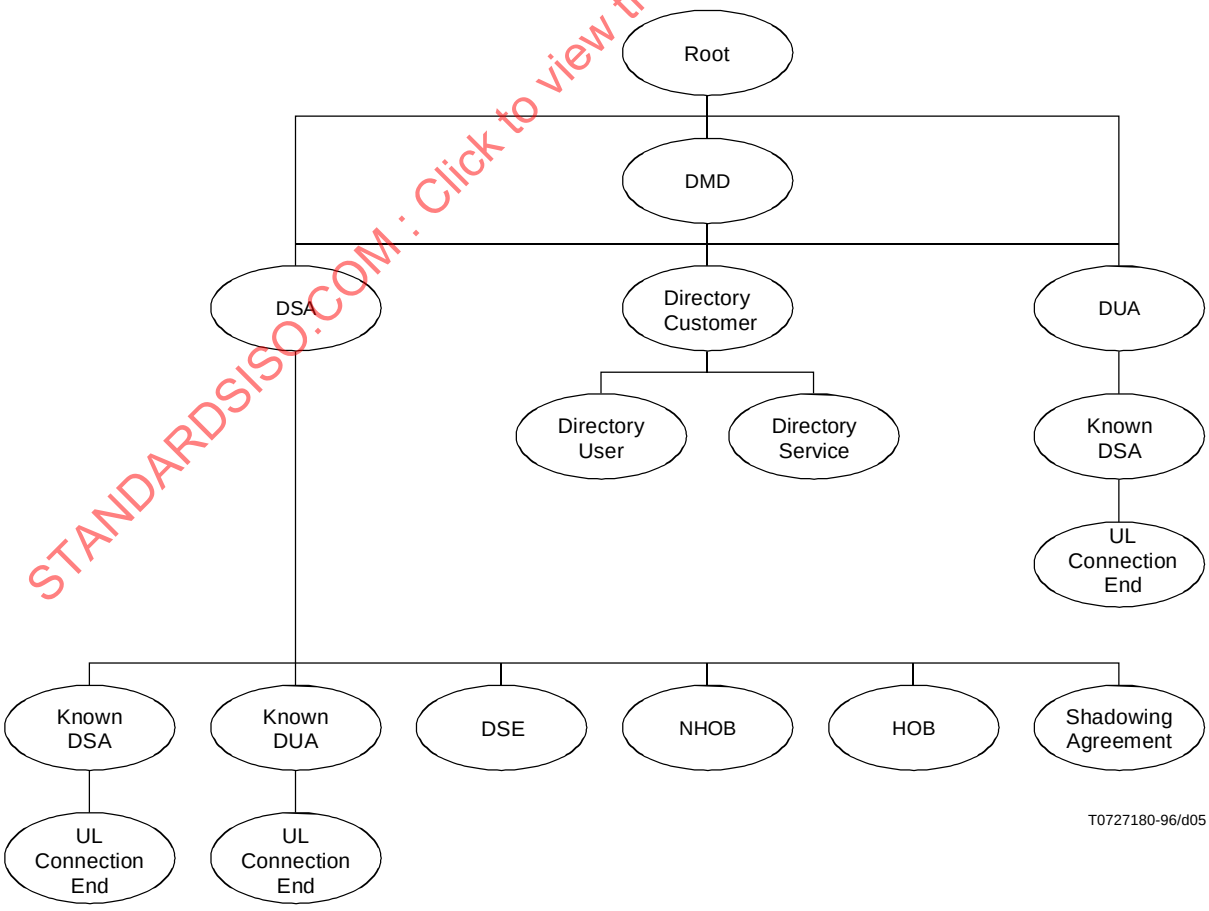

Figure 5 – Directory management containment hierarchy

SECTION 4 – MANAGED OBJECTS

10 Directory managed objects

This clause describes the managed objects that enable Directory components to be managed by Systems Management protocols, in order to meet some of the management requirements which are identified in clause 6.

Throughout these definitions, the arcs in the naming tree are characterised by name bindings, which may have various semantics. This Specification gives optional name bindings so that those who use them may have a standardised approach to naming. This does not preclude users from defining other naming trees for their own purposes.

10.1 DSA managed object

A DSA is represented in the OSI Environment as an application process with an application entity representing its communications capabilities. This Section identifies the managed objects used to represent and manage a DSA, its application entity invocations, application associations and operations.

10.1.1 DSA managed object definitions

A DSA is represented by a DSA managed object instance.

Each DSA managed object is named immediately subordinate to the DMD managed object which represents the DMD of which it is a part.

Each DSA is characterised by a DSA package which includes the following attributes:

- Access Point: The **myAccessPoint** attribute of the root DSE in the DSA. This attribute contains the presentation address, protocol information and AE Title of the DSA.
- Supported Application contexts: The object identifiers of the application contexts supported by the DSA.
- Operational State: The operational state of the DSA.
- Administrative State: The administrative state of the DSA.
- Master Entries: The number of entries mastered by the DSA.
- Copy Entries: The number of entry copies held by the DSA.
- Loops Detected: The number of loops detected by the DSA.
- Security Errors: The number of security errors detected by the DSA.
- Name Errors: The number of name errors detected by the DSA.
- Found Local Entries: The number of target entries found by the DSA.
- Service Errors: The number of Service Errors detected by the DSA.
- Referrals: The number of referrals used by the DSA.
- Alias Dereferences: The number of alias dereferences performed by the DSA.
- Chainings: The number of chained operations initiated by the DSA.
- Invalid References: The number of invalid references reported by the DSA.
- Unable to Proceed: The number of unable to proceed errors reported by the DSA.
- Out of Scope: The number of out of scope errors reported by the DSA.
- No Such Object: The number of no such object errors reported by the DSA.
- Alias Problem: The number of alias problem errors reported by the DSA.
- Alias Dereferencing problem: The number of alias dereferencing problem errors reported by the DSA.
- Affects Multiple DSAs: The number of affects Multiple DSA errors reported by the DSA.
- Unvailable Critical Extension: The number of unavailable critical extension errors reported by the DSA.
- Time Limit Exceeded: The number of time limit exceeded errors reported by the DSA.
- Size Limit Exceeded: The number of size limit exceeded errors reported by the DSA.

- Admin Limit Exceeded: The number of administrative limit exceeded errors reported by the DSA.
- Size Limit: The maximum size limit policy for the DSA. The DSA uses this value as the size limit if the size limit service control exceeds this value or is not included in an operation.
- Time Limit: The maximum time limit policy for the DSA. The DSA uses this value as the time limit if the time limit service control exceeds this value or is not included in an operation.
- Common Name: The naming attribute.
- DSA Scope of Referral: The limitation on the DSA of referrals to one of DMD, country or global scope.
- DSA Scope of Chaining: The limitation on the DSA of chaining to one of DMD, country or global scope.
- Peer Entity Authentication Policy: The types of peer entity authentication supported by the DSA.
- Request Authentication Policy: The types of request authentication supported by the DSA.
- Result Authentication Policy: The types of result authentication supported by the DSA.
- DSP Association Establishment: The directions (inward/outward) of association establishment supported by the DSA for DSP associations.
- DOP Association Establishment: The directions (inward/outward) of association establishment supported by the DSA for DOP associations.
- DISP Association Establishment: The directions (inward/outward) of association establishment supported by the DSA for DISP associations.
- Max DAP Associations: The maximum number of concurrent DAP associations permitted by the DSA.
- Max DSP Associations: The maximum number of concurrent DSP associations permitted by the DSA.
- Max DOP Associations: The maximum number of concurrent DOP associations permitted by the DSA.
- Max DISP Associations: The maximum number of concurrent DISP associations permitted by the DSA.
- DAP Association Timeout: The number of seconds after which a DSA shall timeout a quiescent DAP Association.
- DSP Association Timeout: The number of seconds after which a DSA shall timeout a quiescent DSP Association.
- DOP Association Timeout: The number of seconds after which a DSA shall timeout a quiescent DOP Association.
- DISP Association Timeout: The number of seconds after which a DSA shall timeout a quiescent DISP Association.
- DSA Active Associations Threshold: The total number of active associations supported by the DSA.
- Paged Results Maximum Identifiers: The maximum number of active query references supported by the DSA (on a per association basis).
- Paged Results Expunge Timer in Seconds: The maximum time limit allowed for active query references before they are deleted by the DSA.
- Prohibit Chaining: The chaining policy of the DSA. If this value is true, then the DSA will not chain.

Each DSA is characterised by a DSA package which specifies the following behaviours for the following notifications defined in CCITT Rec. X.721 | ISO/IEC 10165-2:

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":stateChange

- State Change Behaviour: A notification with this behaviour is generated whenever a DSA changes its operational state.

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalViolation

- Name Error Behaviour: A notification with this behaviour is generated whenever a DSA detects a naming problem.
- Service Error Behaviour: A notification with this behaviour is generated whenever the Directory detects a service error.
- Attribute Error Behaviour: A notification with this behaviour is generated whenever the Directory detects an attribute error.
- Update Error Behaviour: A notification with this behaviour is generated whenever the Directory detects an update error.

- Alias Problem Behaviour: A notification with this behaviour is generated whenever the Directory detects an alias problem error.
- Alias Dereferencing Problem Behaviour: A notification with this behaviour is generated whenever the Directory detects an alias dereferencing problem error.

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":processingErrrorAlarm

- Unavailable Critical Extension Behaviour: A notification with this behaviour is generated whenever a DSA is required to use a critical extension that it does not support.
- Unable to Proceed Behaviour: A notification with this behaviour is generated whenever the Directory is unable to proceed with name resolution or operation evaluation.
- Invalid Reference Behaviour: A notification with this behaviour is generated whenever the Directory detects an invalid knowledge reference.
- Loop Detected Behaviour: A notification with this behaviour is generated whenever the Directory detects a loop in the configuration of the Directory distribution.
- Resource Exhausted Behaviour: A notification with this behaviour is generated whenever a DSA runs out of resources.

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":securityServiceOrMechanismViolation

- Authentication Failure Behaviour: A notification with this behaviour is generated whenever an authentication failure is detected by a DSA.
- Access Control Failure Behaviour: A notification with this behaviour is generated whenever a DSA detects an attempt to access an object prohibited by an access control policy.

"ITU-T Rec. X.723 (1993) | ISO/IEC 10165-5:1994":communicationsInformation

- Operation Request Behaviour: A notification with this behaviour is generated whenever a DSA receives a DAP or DSP operation request.
- Operation Response Behaviour: A notification with this behaviour is generated whenever a DSA transmits a DAP or DSP result or error.

10.1.2 Directory service package definitions

This subclause describes the Directory service conditional packages that may be included in a DSA managed object instance. Each Directory service conditional package is included in the DSA managed object class instance if the DSA implements the corresponding abstract service.

10.1.2.1 Read package

Each DSA that supports Read operations is characterised by a managed object package which specifies the following attribute:

- Read Operations Processed: This attribute is used to count the number of Read operations that the DSA has processed in the evaluation phase. For each Read operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.2 Compare package

Each DSA that supports Compare operations is characterised by a managed object package which specifies the following attribute:

- Compare Operations Processed: This attribute is used to count the number of Compare operations that the DSA has processed in the evaluation phase. For each Compare operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.3 Abandon package

Each DSA that supports Abandon operations is characterised by a managed object package which specifies the following attribute:

- Abandon Operations Processed: This attribute is used to count the number of Abandon operations that the DSA has processed. For each Abandon operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.4 Search package

Each DSA that supports Search operations is characterised by a managed object package which specifies the following attributes:

- Search Base Operations Processed: This attribute is used to count the number of Search operations that the DSA has processed which only refer to the base object. For each such Search operation that the DSA evaluates, the DSA increases the counter by 1.
- Search One Level Operations Processed: This attribute is used to count the number of Search operations that the DSA has processed which refer to the base object's immediate subordinates. For each such Search operation that the DSA evaluates, the DSA increases the counter by 1.
- Search Subtree Operations Processed: This attribute is used to count the number of Search operations that the DSA has processed which refer to a whole subtree of the DIT. For each such Search operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.5 List package

Each DSA that supports List operations is characterised by a managed object package which specifies the following attribute:

- List Operations Processed: This attribute is used to count the number of List operations that the DSA has processed in the evaluation phase. For each List operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.6 Add Entry package

Each DSA that supports Add Entry operations is characterised by a managed object package which specifies the following attribute:

- Add Operations Processed: This attribute is used to count the number of Add Entry operations that the DSA has processed in the evaluation phase. For each Add Entry operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.7 Remove Entry package

Each DSA that supports Remove Entry operations is characterised by a managed object package which specifies the following attribute:

- Remove Operations Processed: This attribute is used to count the number of Remove Entry operations that the DSA has processed in the evaluation phase. For each Remove Entry operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.8 Modify Entry package

Each DSA that supports Modify Entry operations is characterised by a managed object package which specifies the following attribute:

- Modify Operations Processed: This attribute is used to count the number of Modify Entry operations that the DSA has processed in the evaluation phase. For each Modify Entry operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.9 Modify DN package

Each DSA that supports Modify DN operations is characterised by a managed object package which specifies the following attributes:

- Modify DN Operations Processed: This attribute is used to count the number of Modify DN operations that the DSA has processed in the evaluation phase. For each Modify DN operation that the DSA evaluates, the DSA increases the counter by 1.
- Modify DN Rename Only Operations Processed: This attribute is used to count the number of Modify DN operations which do not supply a value of **newSuperior** that the DSA has processed in the evaluation phase. For each such operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.10 Chained Read package

Each DSA that supports Chained Read operations is characterised by a managed object package which specifies the following attribute:

– Chained Read Operations Processed: This attribute is used to count the number of Chained Read operations that the DSA has processed in the evaluation phase. For each Chained Read operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.11 Chained Compare package

Each DSA that supports Chained Compare operations is characterised by a managed object package which specifies the following attribute:

– Chained Compare Operations Processed: This attribute is used to count the number of Chained Compare operations that the DSA has processed in the evaluation phase. For each Chained Compare operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.12 Chained Abandon package

Each DSA that supports Chained Abandon operations is characterised by a managed object package which specifies the following attribute:

– Chained Abandon Operations Processed: This attribute is used to count the number of Chained Abandon operations that the DSA has processed. For each Chained Abandon operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.13 Chained Search package

Each DSA that supports Chained Search operations is characterised by a managed object package which specifies the following attributes:

– Chained Search Base Operations Processed: This attribute is used to count the number of Chained Search operations that the DSA has processed which only refer to the base object. For each such Chained Search operation that the DSA evaluates, the DSA increases the counter by 1.

– Chained Search One Level Operations Processed: This attribute is used to count the number of Chained Search operations that the DSA has processed which refer to the base object's immediate subordinates. For each such Chained Search operation that the DSA evaluates, the DSA increases the counter by 1.

– Chained Search Subtree Operations Processed: This attribute is used to count the number of Chained Search operations that the DSA has processed which refer to a whole subtree of the DIT. For each such Chained Search operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.14 Chained List package

Each DSA that supports Chained List operations is characterised by a managed object package which specifies the following attribute:

– Chained List Operations Processed: This attribute is used to count the number of Chained List operations that the DSA has processed in the evaluation phase. For each Chained List operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.15 Chained Add Entry package

Each DSA that supports Chained Add Entry operations is characterised by a managed object package which specifies the following attribute:

– Chained Add Operations Processed: This attribute is used to count the number of Chained Add Entry operations that the DSA has processed in the evaluation phase. For each Chained Add Entry operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.16 Chained Remove Entry package

Each DSA that supports Chained Remove Entry operations is characterised by a managed object package which specifies the following attribute:

– Chained Remove Operations Processed: This attribute is used to count the number of Chained Remove Entry operations that the DSA has processed in the evaluation phase. For each Chained Remove Entry operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.17 Chained Modify Entry package

Each DSA that supports Chained Modify Entry operations is characterised by a managed object package which specifies the following attribute:

- Chained Modify Operations Processed: This attribute is used to count the number of Chained Modify Entry operations that the DSA has processed in the evaluation phase. For each Chained Modify Entry operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.2.18 Chained Modify DN package

Each DSA that supports Chained Modify DN operations is characterised by a managed object package which specifies the following attribute:

- Chained Modify DN Operations Processed: This attribute is used to count the number of Chained Modify DN operations that the DSA has processed in the evaluation phase. For each Chained Modify DN operation that the DSA evaluates, the DSA increases the counter by 1.

10.1.3 DSA Information Tree operational information definitions

This subclause describes the managed objects used to represent and manage the operational information for a DSA's DSA Information Tree (DIB fragment).

10.1.3.1 DSE managed objects

Each DSA Specific Entry is represented by a DSE managed object.

Each DSE managed object is named subordinate to the DSA managed object by using its Directory name as the management relative distinguished name.

Each DSE is characterised by a DSE package which specifies the following attributes:

- Distinguished Name: The Distinguished Name of the DSE.
- Specific Knowledge: The knowledge information of a naming context's immediate superior reference, or a subordinate reference.
- Non-specific Knowledge: The knowledge reference for a non-specific subordinate reference if present in the DSE.
- Administrative Role: The Administrative Role for the DSE if the DSE represents an administrative point.
- Supplier Knowledge: The supplier knowledge reference for a naming context if the naming context is supplied by another DSA.
- Consumer Knowledge: The consumer knowledge reference for a naming context if the naming context is supplied to another DSA.
- Secondary Shadows: The secondary shadows information for a naming context.
- Access Point: The Access Point information for the root DSE.
- DSE Type: The **DSEtype** of the DSE.
- Create Timestamp: The create timestamp of the DSE.
- Modify Timestamp: The modify timestamp of the DSE.
- Creators Name: The Name of the user who created the DSE.
- Modifiers Name: The Name of the user who last modified the DSE if the DSE has been modified since it was created.
- Subtree Specification: The subtree specification governing the applicability of a subentry.
- Aliased Entry Name: The target name for an alias.

10.1.4 NHOB definitions

This subclause describes the managed objects used to represent and manage the operational information for a DSA's non-specific hierarchical operational bindings.

10.1.4.1 NHOB managed objects

Each NHOB is represented by a NHOB managed object.

Each NHOB managed object is named subordinate to the DSA managed object by using the distinguished name of the DSE in which the corresponding NSSR resides.

Each NHOB is characterised by a NHOB package which specifies the following attributes:

- Distinguished Name: The Distinguished Name of the immediate superior entry to the subordinate naming context.
- Agreement ID: The agreement identification of the NHOB.
- Agreement Version: The agreement version of the NHOB.
- Operational State: The NHOB's operational state (e.g. active, inactive).
- Remote Access Point: The access point of the peer DSA.
- Use DOP: A flag that indicates that the DOP protocol is used to maintain the operational binding.
- Role: The role that this DSA performs for this NHOB (either holds superior or subordinate naming context).

Each NHOB is characterised by an NHOB package which specifies the following behaviours for the following notifications defined in CCITT Rec. X.721 | ISO/IEC 10165-2:

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":stateChange

- State Change Behaviour: A notification with this behaviour is generated whenever the state of an NHOB is changed.

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalViolation

- DOP Error: A notification with this behaviour is generated whenever the DOP protocol has detected an error with the operational binding between the DSA and its peer DSA.

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":communicationsInformation

- DOP Complete: A notification with this behaviour is generated whenever the DOP protocol has completed an operation on the NHOB with peer DSA.

10.1.5 HOB definitions

This subclause describes the managed objects used to represent and manage the operational information for a DSA's hierarchical operational bindings.

10.1.5.1 HOB managed objects

Each HOB is represented by a HOB managed object.

Each HOB managed object is named subordinate to the DSA managed object by using the distinguished name of the corresponding DSE.

Each HOB is characterised by a HOB package which specifies the following attributes:

- Distinguished Name: The Distinguished Name of the entry at the root of the subordinate naming context.
- Agreement ID: The agreement identification of the HOB.
- Agreement Version: The agreement version of the HOB.
- Operational State: The HOB's operational state (e.g. active, inactive).
- Peer Access Point: The access point of the peer DSA.
- Use DOP: A flag that indicates that the DOP protocol is used to maintain the operational binding.
- Role: The role that this DSA performs for this HOB (either holds superior or subordinate naming context).

Each HOB is characterised by an HOB package which specifies the following behaviours for the following notifications defined in CCITT Rec. X.721 | ISO/IEC 10165-2:

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":stateChange

- State Change Behaviour: A notification with this behaviour is generated whenever the state of an HOB is changed.

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalViolation

- DOP Error: A notification with this behaviour is generated whenever the DOP protocol has detected an error with the operational binding between the DSA and its peer DSA.

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":communicationsInformation

- DOP Complete: A notification with this behaviour is generated whenever the DOP protocol has completed an operation on the HOB with peer DSA.

10.1.6 Shadowing Agreement Definitions

This subclause describes the managed objects used to represent and manage the operational information for a DSA's shadowing agreements.

10.1.6.1 Shadowing Agreement managed objects

Each shadowing agreement is represented by a Shadowing Agreement managed object.

Each Shadowing Agreement managed object is named subordinate to the DSA managed object by using the Directory name of the naming context containing the unit of replication as the RDN.

Each shadowing agreement is characterised by a Shadowing Agreement package which specifies the following attributes:

- Distinguished Name: The Distinguished Name of the naming context containing the unit of replication.
- Agreement ID: The shadowing agreement identification.
- Agreement Version: The shadowing agreement version.
- Operational State: The shadowing agreement's operational state (e.g. active, inactive).
- Shadow Subject: The unit of replication for the shadowing agreement.
- Update Mode: The update mode for the shadowing agreement (supplier initiated, consumer initiated, on change).
- Master Access Point: The access point of the master, if known.
- Secondary Shadows: The access points of any known secondary shadow consumers.
- Remote Access Point: The access point of the peer DSA.
- Shadowing Role: The role that this DSA performs for this agreement (either supplier or consumer).
- Last Update Time: The recorded time of last update for the shadowing agreement.
- Shadowing Schedule: The update schedule information being operated by the DSA for the shadowing agreement.
- Use DOP: A flag that indicates that the DOP protocol is used to maintain the shadowing agreement.
- Next Update Time: The time when the next update should occur.

Each shadowing agreement is characterised by a shadowing agreement package which specifies the following behaviours for the following notifications defined in CCITT Rec. X.721 | ISO/IEC 10165-2:

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": stateChange

- State Change Behaviour: A notification with this behaviour is generated whenever the state of an shadowing agreement is changed.

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": operationalViolation

DOP Error: A notification with this behaviour is generated whenever the DOP protocol has detected an error with the shadowing agreement between the DSA and its peer DSA.

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":communicationsInformation

- DOP Complete: A notification with this behaviour is generated whenever the DOP protocol has completed an operation on the shadowing agreement with peer DSA.
- Shadow Update Complete Behaviour: A notification with this behaviour is generated whenever a DSA shadowing operation sequence succeeds.
- Shadow Error Behaviour: A notification with this behaviour is generated whenever the Directory detects a shadow error.

Each shadowing agreement is characterised by a shadowing agreement:

- Update Shadow: An action which forces an out-of-band shadow update sequence to be performed.

10.2 Known DSA managed objects

The Known DSA is represented in the OSI Environment as an application process with an application entity representing its communications capabilities. The Known DSA represents a peer DSA application entity with which the local Directory component, either a DUA or a DSA, interacts. This Section identifies the managed objects used to represent and manage the Known DSA, its application entity invocations, application associations and operations.

10.2.1 Known DSA managed object definitions

The Known DSA Managed Object is derived from Communications Entity defined in ITU-T Rec. X.723 | ISO/IEC 10165-5. Each known DSA is characterised by a package which includes the following attributes:

- Remote Access Point: The access point of the peer DSA.
- Supported Application Contexts: The application contexts that the local Directory component knows that the peer DSA supports.
- Credentials: The credentials used by this Directory component to authenticate itself to the peer DSA.
- Reverse Credentials: The credentials used by the peer DSA to authenticate itself to this Directory component.
- Directory Quality of Protection: The quality of protection used between this Directory component and the peer DSA.
- Max Inbound Assocs: The maximum number of BIND requests that the peer DSA supports from this Directory component.
- Max Outbound Assocs: The maximum number of associations this Directory component will accept from the peer DSA, if any.
- Time of Last Attempt: The time when the last BIND attempt was made to the peer DSA.
- Time of Last Success: The time when the last BIND was accepted by the peer DSA.
- Current Active Inbound Assocs: The number of associations between this Directory component and the peer DSA that were initiated by this Directory component.
- Current Active Outbound Assocs: The number of associations between this Directory component and the peer DSA that were initiated by the peer DSA, if any.
- Accum Inbound Assocs: The count of the number of associations initiated by the Directory component to the peer DSA.
- Accum Outbound Assocs: The count of the number of associations initiated by the peer DSA for this Directory component, if any.
- Accum Failed Inbound Assocs: The count of the number of failed associations attempts initiated by the Directory component to the peer DSA.
- Accum Failed Outbound Assocs: The count of the number of failed associations attempts initiated by the peer DSA for this Directory component, if any.
- Request Counter: The total number of requests issued by this Directory component to the peer DSA.
- Reply Counter: The total number of replies received by this Directory component from the peer DSA.
- Requests Failed Counter: The total number of failed requests received by this Directory component from the peer DSA.

Each Known DSA is characterised by a Known DSA package which specifies the following behaviours for the following notifications defined in CCITT Rec. X.721 | ISO/IEC 10165-2:

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":communicationsAlarm

10.3 Known DUA managed objects

The Known DUA is represented in the OSI Environment as an application process with an application entity representing its communications capabilities. The Known DUA represents another DUA application entity with which the local Directory component interacts. This Section identifies the managed objects used to represent and manage the Known DUA, its application entity invocations, application associations and operations.

10.3.1 Known DUA managed object definitions

The Known DUA Managed Object is derived from Communications Entity defined in ITU-T Rec. X.723 | ISO/IEC 10165-5. Each known DUA is characterised by a package which includes the following attributes:

- Remote Access Point: The access point of the DUA.
- Supported Application Contexts: The application contexts that the local Directory component knows that the DUA supports.
- Credentials: The credentials used by this Directory component to authenticate itself to the DUA.
- Reverse Credentials: The credentials used by the DUA to authenticate itself to this Directory component.
- Time of Last Access: The time when the last BIND was accepted by the DSA.
- Current Active Assocs: The number of associations between this Directory component and the DUA.
- Accum Assocs: The count of the total number of associations between this Directory component to the DUA.

Each Known DUA is characterised by a Known DUA package which specifies the following behaviours for the following notifications defined in CCITT Rec. X.721 | ISO/IEC 10165-2:

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":communicationsAlarm

10.4 Upper layer definitions

This subclause specifies the management definitions for the OSI upper layers used by Directory components.

10.4.1 Upper layer connection endpoint managed object class definitions

Each association that is being used by a Directory component is represented by an upper layer connection endpoint managed object. The upper layer connection endpoint managed object class is a subclass of Single Peer Connection defined in ITU-T Rec. X.723 | ISO/IEC 10165-5.

Each upper layer connection endpoint managed object may be named subordinate to either Known DSA or Known DUA by using the association identifier. It would be named subordinate to a Known DSA for associations with that DSA and subordinate to a Known DUA for associations with that DUA.

Each upper layer connection endpoint is characterised by an upper layer connection endpoint package which specifies the following attributes:

- Calling AE Title: The AE title of the peer application entity.
- Application Context In Use: The application context in use for the association.

10.5 DUA managed objects

A DUA is represented in the OSI environment as an application process with the application entities representing the communications capabilities. This Section identifies the managed objects used to represent and manage the DUA.

10.5.1 DUA managed object definitions

A DUA is represented by a DUA Managed Object instance.

Each DUA is characterised by a DUA package which includes the following attributes:

- homeDSA: The name of the Known DSA managed object that represents the DSA to be used by the DUA.
- dUATimeout: The number of seconds of inactivity on the association before the association is aborted.
- subSchema: The subschema definitions to be used by the DUA.

Each DUA is characterised by a DUA package which specifies the following actions:

- Use Remote DSA: An action which, if successful, causes the DUA to use the remote DSA as its Directory service access point.
- Use Home DSA: An action which, if successful, causes the DUA to use the home DSA as its Directory service access point.

10.6 Directory Service managed objects

A Directory service specification for a service is agreed between a Directory service provider and its customers. This Section identifies the managed objects used to represent and manage the Directory services.

10.6.1 Directory Service managed object definitions

A Directory service is represented by a Directory Service Managed Object instance.

Each Directory service is characterised by the following conditional package:

- Directory Information Service package which is present if the DSA allows the Directory service manager to control the information processing capability of the Directory. This package includes the following attributes:
  - serviceIdentifier: The identifications of the service;
  - serviceDescription: A description of the service;
  - allowedDirectoryInformationServiceElement: A list of the permitted operations for the service;
  - disallowedDirectoryInformationServiceElement: A list of the precluded operations for the service;
  - accessor: The list of names of the Directory users that can access the service;
  - TimeLimit: The maximum value of the time limit that is used to provide the service; and
  - SizeLimit: The maximum value of the size limit that is used to provide the service.
- Directory Control Service package which is present if the DSA allows the Directory service manager to manage the operational activity of the Directory. This package includes the following attributes:
  - serviceIdentifier: The identifications of the service;
  - serviceDescription: A description of the service;
  - maxTimeForResults: The maximum time that will be used to provide the service; and
  - maxEntriesReturned: The maximum number of entries that will be returned by an operation providing the service.

10.6.2 Directory Customer managed object definitions

A Directory customer is responsible for procuring directory services from a Directory service provider. A Directory customer is represented by a Directory Customer Managed Object instance.

Each Directory customer is characterised by a Directory Customer Package which contains the following attributes:

- Directory Customer Name: The Directory Name of the customer.
- Directory Customer Address: The address of the customer.

10.6.3 Directory User managed object definitions

A Directory user is a user of Directory services. A Directory user is represented by a Directory User Managed Object.

Each Directory user is characterised by a Directory User Package which contains the following attribute:

- Directory User Name: The name of the Directory user.

10.7 Directory Management Domain managed objects

A DMD is represented by a DMD Managed Object.

Each DMD is characterised by a DMD Package which contains the following attribute:

- DMD Name: The name of the DMD.

Annex A

Managed object definitions

(This annex forms an integral part of this Recommendation | International Standard)

This annex contains proposed managed object definitions that enable a DSA to be managed by Systems Management protocols according to the model described in clause 9 and the descriptions of managed objects in clause 10.

A.1 Management of a DSA

A DSA is represented in the OSI Environment as an application process with application entities representing its communications capabilities. This Section identifies the managed objects used to represent and manage a DSA, its application entity invocations, application associations and operations.

A.1.1 DSA managed object definitions

The following definition specifies the DSA managed objects used to represent a DSA in an end system.

```
dSA MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":top ;
  CHARACTERIZED BY dSAPackage ;
  CONDITIONAL PACKAGES
    readPackage PRESENT IF 'the DSA supports the read operation',
    comparePackage PRESENT IF 'the DSA supports the compare operation',
    abandonPackage PRESENT IF 'the DSA supports the abandon operation',
    searchPackage PRESENT IF 'the DSA supports the search operation',
    listPackage PRESENT IF 'the DSA supports the list operation',
    addEntryPackage PRESENT IF 'the DSA supports the addEntry operation',
    removeEntryPackage PRESENT IF 'the DSA supports the removeEntry operation',
    modifyEntryPackage PRESENT IF 'the DSA supports the modifyEntry operation',
    modifyDNPackage PRESENT IF 'the DSA supports the modifyDN operation',
    chainedReadPackage PRESENT IF 'the DSA supports the read operation',
    chainedComparePackage PRESENT IF 'the DSA supports the compare operation',
    chainedAbandonPackage PRESENT IF 'the DSA supports the abandon operation',
    chainedSearchPackage PRESENT IF 'the DSA supports the search operation',
    chainedListPackage PRESENT IF 'the DSA supports the list operation',
    chainedAddEntryPackage PRESENT IF 'the DSA supports the addEntry operation',
    chainedRemoveEntryPackage PRESENT IF 'the DSA supports the removeEntry operation',
    chainedModifyEntryPackage PRESENT IF 'the DSA supports the modifyEntry operation',
    chainedModifyDNPackage PRESENT IF 'the DSA supports the modifyDN operation';
  REGISTERED AS {DirectoryManagementModule.id-moc-dsa} ;
```

A.1.1.1 DSA name binding definitions

The following definition specifies the naming relationship of DSAs to other managed objects. DSAs are named subordinate to a DMD.

```
dSANB NAME BINDING
    SUBORDINATE OBJECT CLASS dSA AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS dMD AND SUBCLASSES;
    WITH ATTRIBUTE dirCommonName ;
    CREATE WITH-REFERENCE-OBJECT ;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-dsa-name-binding} ;
```

A.1.1.2 DSA package definition

The following definition specifies the package for DSAs.

```
dSAPackage PACKAGE
    BEHAVIOUR dSABehaviour BEHAVIOUR
```

DEFINED AS ! This package contains the definitions that manage the DSA itself ! ;,
nameErrorNotificationBehaviour,
serviceErrorNotificationBehaviour,
attributeErrorNotificationBehaviour,
updateErrorNotificationBehaviour,
shadowErrorNotificationBehaviour,
unavailableCriticalExtensionNotificationBehaviour,
resourceExhaustedNotificationBehaviour,
authenticationFailureNotificationBehaviour,
accessControlFailureNotificationBehaviour,
aliasProblemNotificationBehaviour,
aliasDereferencingProblemNotificationBehaviour,
unableToProceedNotificationBehaviour,
invalidReferenceNotificationBehaviour,
loopDetectedNotificationBehaviour,
operationRequestNotificationBehaviour,
operationResponseNotificationBehaviour,
shadowUpdateCompleteNotificationBehaviour;
ATTRIBUTES
dirCommonName GET ,
"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalState GET,
"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":administrativeState GET-REPLACE,
accessPoint GET-REPLACE ,
masterEntries REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
copyEntries REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
loopsDetected REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
securityErrors REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
nameErrors REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
foundLocalEntries REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
referrals REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET,
serviceErrors REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
aliasDereferences REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
chainings REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
invalidReferences REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
unableToProceed REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
outOfScope REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
noSuchObject REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
aliasProblem REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
aliasDereferencingProblem REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
affectsMultipleDSAs REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
unavailableCriticalExtension REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
timeLimitExceeded REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
sizeLimitExceeded REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
adminLimitExceeded REPLACE-WITH-DEFAULT DEFAULT VALUE
DirectoryManagementModule.zero GET ,
sizeLimit GET-REPLACE ,
timeLimit GET-REPLACE ,
prohibitChaining GET-REPLACE ,
dSAScopeOfReferral GET-REPLACE ,

```
        dSAScopeOfChaining GET-REPLACE ,
        peerEntityAuthenticationPolicy GET-REPLACE ,
        requestAuthenticationPolicy GET-REPLACE ,
        resultAuthenticationPolicy GET-REPLACE ,
        dSPAssociationEstablishment GET-REPLACE ,
        dOPAssociationEstablishment GET-REPLACE ,
        dISPAssociationEstablishment GET-REPLACE ,
        maxDAPAssociations GET-REPLACE ,
        maxDSPAssociations GET-REPLACE ,
        maxDOPAssociations GET-REPLACE ,
        maxDISPAssociations GET-REPLACE ,
        dAPAssociationTimeout GET-REPLACE ,
        dSPAssociationTimeout GET-REPLACE ,
        dOPAssociationTimeout GET-REPLACE ,
        dISPAssociationTimeout GET-REPLACE ,
        dSAActiveAssociationsThreshold GET-REPLACE ,
        pagedResultsMaximumIdentifiers GET-REPLACE ,
        pagedResultsExpungeTimerInSeconds GET-REPLACE ,
        supportedApplicationContexts GET-REPLACE ADD-REMOVE;
    NOTIFICATIONS
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":stateChange,
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalViolation
            entryName
            nameProblem
            traceInformation
            serviceProblem
            operation
            aliasedRDN
            aliasDereferenced
            attributeProblem
            attributeType
            attributeValue,
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":processingErrorAlarm
            entryName
            operation
            extensions
            resource,
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":
          securityServiceOrMechanismViolation
            entryName
            authenReason
            operation,
        "ITU-T Rec. X.723 (1993) | ISO/IEC 10165-5:1994":communicationsInformation
            operationIdentifier
            operationIdentifierDN
            pDU;
    REGISTERED AS {DirectoryManagementModule.id-mp-dsaPackage} ;
```

A.1.1.3 DSA notification parameters

The following parameter definitions are used with the notifications for DSAs.

```
nameProblem PARAMETER
    CONTEXT EVENT-INFO;
    WITH SYNTAX DirectoryManagementModule.NameProblem ;
    BEHAVIOUR nameProblemBehaviour BEHAVIOUR
        DEFINED AS Reason why a nameError has been detected by the DSA ;;
    REGISTERED AS {DirectoryManagementModule.id-mpa-nameProblem};

traceInformation PARAMETER
    CONTEXT EVENT-INFO;
    WITH SYNTAX DirectoryManagementModule.TraceInformation ;
    BEHAVIOUR traceInfo-B BEHAVIOUR
        DEFINED AS The trace information associated with the operation ;;
    REGISTERED AS {DirectoryManagementModule.id-mpa-traceInformation};

serviceProblem PARAMETER
    CONTEXT EVENT-INFO;
    WITH SYNTAX DirectoryManagementModule.ServiceProblem ;
```

```
        BEHAVIOUR serviceProblemBehaviour BEHAVIOUR
            DEFINED AS Reason why a serviceError has been detected by the DSA ;;
        REGISTERED AS {DirectoryManagementModule.id-mpa-serviceProblem};

entryName PARAMETER
        CONTEXT EVENT-INFO;
        WITH SYNTAX DirectoryManagementModule.Name ;
        BEHAVIOUR entryNameBehaviour BEHAVIOUR
            DEFINED AS The name of the entry associated with the operation that caused the
                notifications ;;
        REGISTERED AS {DirectoryManagementModule.id-mpa-entryName};

operation PARAMETER
        CONTEXT EVENT-INFO;
        WITH SYNTAX DirectoryManagementModule.MgtInteger ;
        BEHAVIOUR operationBehaviourBehaviour BEHAVIOUR
            DEFINED AS The operation code that caused the notification to be generated by the
                DSA ;;
        REGISTERED AS {DirectoryManagementModule.id-mpa-operation};

attributeProblem PARAMETER
        CONTEXT EVENT-INFO;
        WITH SYNTAX DirectoryManagementModule.AttributeProblem ;
        BEHAVIOUR attributeProblemBehaviour BEHAVIOUR
            DEFINED AS Reason why an attributeError has been detected by the DSA ;;
        REGISTERED AS {DirectoryManagementModule.id-mpa-attributeProblem};

attributeType PARAMETER
        CONTEXT EVENT-INFO;
        WITH SYNTAX DirectoryManagementModule.AttributeType ;
        BEHAVIOUR attributeTypeBehaviour BEHAVIOUR
            DEFINED AS The attribute type in error ;;
        REGISTERED AS {DirectoryManagementModule.id-mpa-attributeType};

attributeValue PARAMETER
        CONTEXT EVENT-INFO;
        WITH SYNTAX DirectoryManagementModule.AttributeValue;
        BEHAVIOUR attributeValueBehaviour BEHAVIOUR
            DEFINED AS The attribute value in error ;;
        REGISTERED AS {DirectoryManagementModule.id-mpa-attributeValue};

resource PARAMETER
        CONTEXT EVENT-INFO;
        WITH SYNTAX DirectoryManagementModule.ResourceSyntax ;
        BEHAVIOUR resourceBehaviour BEHAVIOUR
            DEFINED AS An identification of the resource that has become exhausted ;;
        REGISTERED AS {DirectoryManagementModule.id-mpa-resource};

authenReason PARAMETER
        CONTEXT EVENT-INFO;
        WITH SYNTAX DirectoryManagementModule.AuthenReasonSyntax ;
        BEHAVIOUR authenReasonBehaviour BEHAVIOUR
            DEFINED AS The reason why the authentications failed ;;
        REGISTERED AS {DirectoryManagementModule.id-mpa-authenReason};

extensions PARAMETER
        CONTEXT EVENT-INFO;
        WITH SYNTAX DirectoryManagementModule.MgtBitString ;
        BEHAVIOUR extensionsBehaviour BEHAVIOUR
            DEFINED AS The critical extensions not supported by the DSA ;;
        REGISTERED AS {DirectoryManagementModule.id-mpa-extensions};

aliasedRDNs PARAMETER
        CONTEXT EVENT-INFO;
        WITH SYNTAX DirectoryManagementModule.RDNSequence ;
        BEHAVIOUR aliasedRDNsBehaviour BEHAVIOUR
            DEFINED AS The aliased RDNs processed by the DSA ;;
        REGISTERED AS {DirectoryManagementModule.id-mpa-aliasedRDNs};

aliasDereferenced PARAMETER
        CONTEXT EVENT-INFO;
        WITH SYNTAX DirectoryManagementModule.Name ;
```

BEHAVIOUR aliaseDereferencedBehaviour BEHAVIOUR
DEFINED AS The name of the dereferenced alias ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-aliasDereferenced};

referenceType PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX DirectoryManagementModule.ReferenceType ;
BEHAVIOUR referenceTypeBehaviour BEHAVIOUR
DEFINED AS The reference type of the knowledge reference ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-referenceType};

operationProgress PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX DirectoryManagementModule.OperationProgress ;
BEHAVIOUR operationProgressBehaviour BEHAVIOUR
DEFINED AS The operation progress when the error was detected ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-operationProgress};

pDU PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX DirectoryManagementModule.MgtOctetString ;
BEHAVIOUR pDUBehaviour BEHAVIOUR
DEFINED AS The octets of a PDU sent or received by the entity ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-pDU};

operationIdentifier PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX DirectoryManagementModule.MgtInteger ;
BEHAVIOUR operationIdentifierBehaviour BEHAVIOUR
DEFINED AS The operation identifier for the operation of response ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-opId};

operationIdentifierDN PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX DirectoryManagementModule.Name ;
BEHAVIOUR operationIdentifierDNBehaviour BEHAVIOUR
DEFINED AS The distinguished name qualifying the operation identifier for the operation of response ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-opIdDN};

A.1.1.4 DSA notification behaviours

The following behaviour definitions are used with the notifications for DSAs.

nameErrorNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects a name error that it reports to the peer.

The **serviceUser** field of the **operationViolation** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **operationViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **operationViolation** notification contains any additional textual information to be conveyed in the notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **nameProblem** parameter indicates the problems that was detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **traceInformation** parameter holds the trace information from a chained operation and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification. ! ;

serviceErrorNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects a service error while processing an operation either as part of the name resolution phase, or as part of the operation evaluation phase.

The **serviceUser** field of the **operationViolation** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **operationViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **operationViolation** notification contains any additional textual information to be conveyed in the notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **serviceProblem** parameter contains an indication of the service error that was detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **operation** parameter contains an indication of the operation that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **traceInformation** parameter holds the trace information from a chained operation and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification. ! ;

attributeErrorNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects an attribute error while processing an operation either as part of the name resolution phase, or as part of the operation evaluation phase.

The **serviceUser** field of the **operationViolation** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **operationViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **operationViolation** notification contains any additional textual information to be conveyed in the notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **attributeProblem** parameter contains an indication of the attribute error that was detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **attributeType** parameter contains the object identifier of the attribute that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **attributeValue** parameter contains the attribute value that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **traceInformation** parameter holds the trace information from a chained operation and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification. ! ;

updateErrorNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects a update error while processing an operation either as part of the name resolution phase, or as part of the operation evaluation phase.

The **serviceUser** field of the **operationViolation** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **operationViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **operationViolation** notification contains any additional textual information to be conveyed in the notification.

The **operation** parameter contains an indication of the operation that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **updateProblem** parameter contains an indication of the update error that was detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **traceInformation** parameter holds the trace information from a chained operation and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification. ! ;

aliasProblemNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects an alias problem.

The **serviceUser** field of the **operationViolation** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **operationViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **operationViolation** notification contains any additional textual information to be conveyed in the notification.

The **operation** parameter contains an indication of the operation that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **aliasedRDNs** parameter contains the aliased RDNs if available and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **aliasDereferenced** parameter contains the name of a dereferenced alias if available and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **traceInformation** parameter holds the trace information from a chained operation and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification. ! ;

aliasDereferencingProblemNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects an alias dereferencing problem.

The **serviceUser** field of the **operationViolation** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **operationViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **operationViolation** notification contains any additional textual information to be conveyed in the notification.

The **operation** parameter contains an indication of the operation that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **aliasedRDNs** parameter contains the aliased RDNs if available and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **aliasDereferenced** parameter contains the name of a dereferenced alias if available and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **traceInformation** parameter holds the trace information from a chained operation and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification. ! ;

unavailableCriticalExtensionNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects a set extension bit it does not understand. This represents an unknown/unimplemented critical extension.

The **serviceUser** field of the **processingErrorAlarm** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **processingErrorAlarm** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **processingErrorAlarm** notification contains any additional textual information to be conveyed in the notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **operation** parameter contains an indication of the operation that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **extensions** parameter contains the critical extension bits that are unknown to the DSA and that are set in the requested operation and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification. ! ;

unableToProceedNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA is unable to proceed with name resolution or during operation evaluation.

The **serviceUser** field of the **processingErrorAlarm** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **processingErrorAlarm** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **processingErrorAlarm** notification contains any additional textual information to be conveyed in the notification.

The **operation** parameter contains an indication of the operation that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **referenceType** parameter contains the reference type of the knowledge reference and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **operationProgress** parameter contains the operation progress information at the time that the error was detected and is conveyed as a parameter in the **processingErrorAlarm** field of the **operationViolation** notification.

The **traceInformation** parameter holds the trace information from a chained operation and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification. ! ;

invalidReferenceNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects an invalid reference.

The **serviceUser** field of the **processingErrorAlarm** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **processingErrorAlarm** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **processingErrorAlarm** notification contains any additional textual information to be conveyed in the notification.

The **operation** parameter contains an indication of the operation that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **referenceType** parameter contains the reference type of the knowledge reference and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **operationProgress** parameter contains the operation progress information at the time that the error was detected and is conveyed as a parameter in the **processingErrorAlarm** field of the **operationViolation** notification.

The **traceInformation** parameter holds the trace information from a chained operation and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification. ! ;

loopDetectedNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects a loop in the configuration of the Directory distribution.

The **serviceUser** field of the **processingErrorAlarm** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **processingErrorAlarm** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **processingErrorAlarm** notification contains any additional textual information to be conveyed in the notification.

The **operation** parameter contains an indication of the operation that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **referenceType** parameter contains the reference type of the knowledge reference and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **traceInformation** parameter holds the trace information from a chained operation and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification. ! ;

resourceExhaustedNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects an exhausted resource.

The **serviceUser** field of the **processingErrorAlarm** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **processingErrorAlarm** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **processingErrorAlarm** notification contains any additional textual information to be conveyed in the notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **operation** parameter contains an indication of the operation that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification.

The **resource** parameter contains an indication of the resource that is exhausted and is conveyed as a parameter in the **additionalInformation** field of the **processingErrorAlarm** notification. ! ;

authenticationFailureNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever an authentication failure occurs.

The **serviceUser** field of the **securityServiceOrMechanismViolation** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **securityServiceOrMechanismViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **securityServiceOrMechanismViolation** notification contains any additional textual information to be conveyed in the notification.

The **authenReason** parameter contains an indication of the reason why the authentication failed and is conveyed as a parameter in the **additionalInformation** field of the **securityServiceOrMechanismViolation** notification. ! ;

accessControlFailureNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects an attempt to access an object prohibited by an access control, policy.

The **serviceUser** field of the **securityServiceOrMechanismViolation** notification contains the authenticated name of the user requesting the operation, or the peer DUA's AE-title.

The **serviceProvider** field of the **securityServiceOrMechanismViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **securityServiceOrMechanismViolation** notification contains any additional textual information to be conveyed in the notification.

The **entryName** parameter contains the Name of the base object for the entry and is conveyed as a parameter in the **additionalInformation** field of the **securityServiceOrMechanism-Violation** notification.

The operation parameter contains an indication of the operation that caused the error and is conveyed as a parameter in the **additionalInformation** field of the **securityServiceOr-MechanismViolation** notification. ! ;

operationRequestNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA received an operation.

The **serviceUser** field of the **communicationsInformation** notification contains the authenticated name of the DSA requesting the operation, or the peer DSA's AE-title.

The **serviceProvider** field of the **communicationsInformation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **communicationsInformation** notification contains any additional textual information to be conveyed in the notification.

The **pDU** parameter contains the PDU received by the DSA for processing and is conveyed as a parameter in the **additionalInformation** field of the **communicationsInformation** notification.

The **operationIdentifier** parameter contains the operation identification of the operation and is conveyed as a parameter in the **additionalInformation** field of the **communications-Information** notification.

The **operationIdentifierDN** parameter contains the distinguished name qualifying the operation identifier and is conveyed as a parameter in the **additionalInformation** field of the **communicationsInformation** notification. For DAP operations, this is the distinguished name of the receiving DSA. For DSP operations, this is the distinguished name of the first element of the received trace information. ! ;

operationResponseNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA sends an operation response (including results and errors).

The **serviceUser** field of the **communicationsInformation** notification contains the authenticated name of the DSA requesting the operation, or the peer DSA's AE-title.

The **serviceProvider** field of the **communicationsInformation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **communicationsInformation** notification contains any additional textual information to be conveyed in the notification.

The **pDU** parameter contains the PDU being sent by the DSA and is conveyed as a parameter in the **additionalInformation** field of the **communicationsInformation** notification.

The **operationIdentifier** parameter contains the operation identification of the operation for which this is the response and is conveyed as a parameter in the **additionalInformation** field of the **communicationsInformation** notification.

The **operationIdentifierDN** parameter contains the distinguished name qualifying the operation identifier and is conveyed as a parameter in the **additionalInformation** field of the **communicationsInformation** notification. For DAP operations, this is the distinguished name of the DSA. For DSP operations, this is the distinguished name of the first element of the received trace information from the corresponding request. ! ;

A.1.2 Directory Service Packages Management Package definitions

The following definitions specify the systems management packages for the Directory Service Packages that may be included in the **dSA** managed object instance.

```
readPackage PACKAGE
    BEHAVIOUR readPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                     access to the Directory read operation. ! ;;
        ATTRIBUTES
            readOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-readPackage} ;

comparePackage PACKAGE
    BEHAVIOUR comparePackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                     access to the Directory compare operation. ! ;;
        ATTRIBUTES
            compareOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-comparePackage} ;

    abandonPackage PACKAGE
    BEHAVIOUR abandonPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                     access to the Directory Abandon operation. ! ;;
        ATTRIBUTES
            abandonOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-abandonPackage} ;

listPackage PACKAGE
    BEHAVIOUR listPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                     access to the Directory list operation. ! ;;
        ATTRIBUTES
            listOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-listPackage} ;

searchPackage PACKAGE
    BEHAVIOUR searchPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                     access to the Directory search operation. ! ;;
        ATTRIBUTES
            searchBaseOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ,
            search1LevelOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ,
            searchSubtreeOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-searchPackage} ;

addEntryPackage PACKAGE
    BEHAVIOUR addEntryPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                     access to the Directory addEntry operation. ! ;;
        ATTRIBUTES
            addEntryOperationsProcessed REPLACE-WITH-DEFAULT
```

```
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-addPackage} ;

removeEntryPackage PACKAGE
    BEHAVIOUR removeEntryPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                    access to the Directory removeEntry operation. ! ;;
        ATTRIBUTES
            removeEntryOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-removePackage} ;

modifyEntryPackage PACKAGE
    BEHAVIOUR modifyEntryPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                    access to the Directory modifyEntry operation. ! ;;
        ATTRIBUTES
            modifyEntryOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-modifyPackage} ;

modifyDNPackage PACKAGE
    BEHAVIOUR modifyDNPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                    access to the Directory modifyDN operation. ! ;;
        ATTRIBUTES
            modifyDNOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ,
            modifyDNRenameOnlyOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-modifyDNPackage} ;

chainedReadPackage PACKAGE
    BEHAVIOUR chainedReadPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                    access to the Directory chainedRead operation. ! ;;
        ATTRIBUTES
            chainedReadOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-chainedReadPackage} ;

chainedComparePackage PACKAGE
    BEHAVIOUR chainedComparePackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                    access to the Directory chainedCompare operation. ! ;;
        ATTRIBUTES
            chainedCompareOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-chainedComparePackage} ;

chainedAbandonPackage PACKAGE
    BEHAVIOUR chainedAbandonPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                    access to the Directory chainedAbandon operation. ! ;;
        ATTRIBUTES
            chainedAbandonOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-chainedAbandonPackage} ;

chainedListPackage PACKAGE
    BEHAVIOUR chainedListPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                    access to the Directory chainedList operation. ! ;;
        ATTRIBUTES
            chainedListOperationsProcessed REPLACE-WITH-DEFAULT
                DEFAULT VALUE DirectoryManagementModule.zero GET ;
        REGISTERED AS {DirectoryManagementModule.id-mp-chainedListPackage} ;

chainedSearchPackage PACKAGE
    BEHAVIOUR chainedSearchPackagebehaviour BEHAVIOUR
        DEFINED AS ! This package holds information about and provides management
                    access to the Directory chainedSearch operation. ! ;;
```

```
            ATTRIBUTES
                chainedSearchBaseOperationsProcessed REPLACE-WITH-DEFAULT
                    DEFAULT VALUE DirectoryManagementModule.zero GET ,
                chainedSearch1LevelOperationsProcessed REPLACE-WITH-DEFAULT
                    DEFAULT VALUE DirectoryManagementModule.zero GET ,
                chainedSearchSubtreeOperationsProcessed REPLACE-WITH-DEFAULT
                    DEFAULT VALUE DirectoryManagementModule.zero GET ;
            REGISTERED AS {DirectoryManagementModule.id-mp-chainedSearchPackage} ;

    chainedAddEntryPackage PACKAGE
        BEHAVIOUR chainedAddEntryPackagebehaviour BEHAVIOUR
            DEFINED AS ! This package holds information about and provides management
                        access to the Directory chainedAddEntry operation. ! ;;
            ATTRIBUTES
                chainedAddENTRYOperationsProcessed REPLACE-WITH-DEFAULT
                    DEFAULT VALUE DirectoryManagementModule.zero GET ;
            REGISTERED AS {DirectoryManagementModule.id-mp-chainedAddPackage} ;

    chainedRemoveEntryPackage PACKAGE
        BEHAVIOUR chainedEntryRemovePackagebehaviour BEHAVIOUR
            DEFINED AS ! This package holds information about and provides management
                        access to the Directory chainedRemoveEntry operation. ! ;;
            ATTRIBUTES
                chainedRemoveEntryOperationsProcessed REPLACE-WITH-DEFAULT
                    DEFAULT VALUE DirectoryManagementModule.zero GET ;
            REGISTERED AS {DirectoryManagementModule.id-mp-chainedRemovePackage} ;

    chainedModifyEntryPackage PACKAGE
        BEHAVIOUR chainedModifyEntryPackagebehaviour BEHAVIOUR
            DEFINED AS ! This package holds information about and provides management
                        access to the Directory chainedModifyEntry operation. ! ;;
            ATTRIBUTES
                chainedModifyEntryOperationsProcessed REPLACE-WITH-DEFAULT
                    DEFAULT VALUE DirectoryManagementModule.zero GET ;
            REGISTERED AS {DirectoryManagementModule.id-mp-chainedModifyPackage} ;

    chainedModifyDNPackage PACKAGE
        BEHAVIOUR chainedModifyDNPackagebehaviour BEHAVIOUR
            DEFINED AS ! This package holds information about and provides management
                        access to the Directory chainedModifyDN operation. ! ;;
            ATTRIBUTES
                chainedModifyDNOperationsProcessed REPLACE-WITH-DEFAULT
                    DEFAULT VALUE DirectoryManagementModule.zero GET ;
            REGISTERED AS {DirectoryManagementModule.id-mp-chainedModifyDNPackage} ;
```

A.1.3 DSA Information Tree operational information definitions

This subclause specifies the management definitions for a DSA's DSA Information Tree operational information.

A.1.3.1 DSA Information Tree managed object class definition

The following definition specifies the DSE managed objects that may be created to manage the necessary aspects of a DSA Information Tree. Each DSE in the DSA Information Tree may be created by the DSA as a managed object instance subordinate to the DSA managed object instance.

```
    dseMO MANAGED OBJECT CLASS
        -- These managed object instances contain the name and operational information for each
        -- Directory managed entry in a naming context held in a DSA.
        DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":top ;
        CHARACTERIZED BY dsePackage ;
        REGISTERED AS {DirectoryManagementModule.id-moc-dse} ;
```

A.1.3.2 DSA Information Tree name binding definition

The following definition specifies the name binding for the DSE managed objects that may be created to represent DSA Information Tree operational information in a DSA.

```
dseNB NAME BINDING
    SUBORDINATE OBJECT CLASS dseMO AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS dSA AND SUBCLASSES;
    WITH ATTRIBUTE distinguishedName;
    BEHAVIOUR
        dseNBBehaviour BEHAVIOUR
            DEFINED AS ! Each DSE in a DSA Information Tree is named by the sequence
                    of RDNs forming its Distinguished Name ! ;;
    CREATE WITH-REFERENCE-OBJECT;
    DELETE ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-dse-name-binding} ;
```

A.1.3.3 DSE package definition

The following definition specifies the packages for the DSE managed objects.

```
dsePackage PACKAGE
    BEHAVIOUR dsePackageBehaviour BEHAVIOUR
        DEFINED AS !The information and actions permitted for managing DSEs. ! ;;
    ATTRIBUTES
        distinguishedName GET ,
        specificKnowledge GET-REPLACE ADD-REMOVE,
        nonSpecificKnowledge GET-REPLACE ADD-REMOVE,
        administrativeRole GET-REPLACE ,
        dseType GET-REPLACE ,
        supplierKnowledge GET-REPLACE ,
        consumerKnowledge GET-REPLACE ,
        secondaryShadows GET-REPLACE ADD-REMOVE,
        createTimestamp GET-REPLACE ,
        modifyTimestamp GET-REPLACE ,
        creatorsName GET-REPLACE ,
        modifiersName GET-REPLACE ,
        aliasedEntryName GET-REPLACE ,
        subtreeSpecification GET-REPLACE ,
        accessPoint GET-REPLACE ;
    REGISTERED AS {DirectoryManagementModule.id-mp-dsePackage} ;
```

A.1.4 NHOB managed object definitions

This subclause specifies the management definitions for a DSA's NHOBs.

A.1.4.1 NHOB managed object class definition

The following definition specifies the managed objects used to represent an NHOB held by a DSA.

```
nHOBMO MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":top ;
    CHARACTERIZED BY nHOBPackage ;
    REGISTERED AS {DirectoryManagementModule.id-moc-nHOBMO} ;
```

A.1.4.2 NHOB name binding definition

The following definition specifies the name binding for the NHOB managed objects that may be created to represent the NHOBs of a DSA.

```
nHOBNB NAME BINDING
    SUBORDINATE OBJECT CLASS nHOBMO AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS dSA AND SUBCLASSES;
    WITH ATTRIBUTE distinguishedName ;
    BEHAVIOUR
        nHOBNBBehaviour BEHAVIOUR
            DEFINED AS ! Each NHOB held by a DSA is named by the sequence of RDNs
                    forming the Distinguished Name of the immediate superior entry
                    of the subordinate naming context ! ;;
    CREATE WITH-REFERENCE-OBJECT ;
    DELETE ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-nHOB-name-binding} ;
```

A.1.4.3 NHOB package definition

The following definition specifies the package for the NHOB managed object.

```
nHOBPackage PACKAGE
    BEHAVIOUR nHOBPackageBehaviour BEHAVIOUR
        DEFINED AS ! The information and actions permitted for managing NHOBs ! ;;
    ATTRIBUTES
        distinguishedName GET,
        agreementID GET,
        agreementVersion GET,
        useDOP GET-REPLACE,
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalState GET,
        remoteAccessPoint GET-REPLACE,
        hOBRole GET ;
    NOTIFICATIONS
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":stateChange,
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalViolation
            operationalBindingID
            dOPProblem,
        "ITU-T Rec. X.723 (1993) | ISO/IEC 10165-5:1994":communicationsInformation
            operationalBindingID;
    REGISTERED AS {DirectoryManagementModule.id-mp-nHOBPackage} ;
```

A.1.4.4 NHOB notification parameters

The following parameter definitions are used with the notifications for NHOBs:

```
operationalBindingID PARAMETER
    CONTEXT EVENT-INFO;
    WITH SYNTAX DirectoryManagementModule.OperationalBindingID ;
    BEHAVIOUR operationalBindingIDBehaviour BEHAVIOUR
        DEFINED AS The operational binding ID associated with the notification ;;
    REGISTERED AS { DirectoryManagementModule.id-mpa-nhob-bind-id} ;

dOPProblem PARAMETER
    CONTEXT EVENT-INFO;
    WITH SYNTAX DirectoryManagementModule.OpBindingErrorParam;
    BEHAVIOUR dOPProblemBehaviour BEHAVIOUR
        DEFINED AS Reason why a DOP Error has been detected by the DSA ;;
    REGISTERED AS {DirectoryManagementModule.id-mpa-mhob-dop-prob} ;
```

A.1.4.5 NHOB notification behaviours

The following behaviour definitions are used with the notifications for NHOBs:

```
dOPErrorNotificationBehaviour BEHAVIOUR
    DEFINED AS
```

! Notifications with this behaviour are generated whenever the DSA detects a DOP error while maintaining the NHOB.

The **serviceUser** field of the **operationViolation** notification contains the authenticated name of the DSA requesting the operation, or the peer DSA's AE-title.

The **serviceProvider** field of the **operationViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **operationViolation** notification contains any additional textual information to be conveyed in the notification.

The **operationalBindingID** parameter contains the operation binding ID of the NHOB for which the error has been detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **DOPProblem** parameter contains an indication of the DOP error that was detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification. ! ;

dOPCompleteNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA successfully completes a DOP operation.

The **serviceUser** field of the **communicationsInformation** notification contains the authenticated name of the DSA requesting the operation, or the peer DSA's AE-title.

The **serviceProvider** field of the **communicationsInformation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **communicationsInformation** notification contains any additional textual information to be conveyed in the notification.

The **operationalBindingID** parameter contains the operational binding ID of the NHOB for which the operation has been completed and is conveyed as a parameter in the **additionalInformation** field of the **communicationsInformation** notification. ! ;

A.1.5 HOB managed object definitions

This subclause specifies the management definitions for a DSA's HOBs.

A.1.5.1 HOB managed object class definition

The following definition specifies the managed objects used to represent a HOB held by a DSA.

```
hOBMO MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":top ;
    CHARACTERIZED BY hOBPackage ;
    REGISTERED AS {DirectoryManagementModule.id-moc-hOBMO} ;
```

A.1.5.2 HOB name binding definition

The following definition specifies the name binding for the HOB managed objects that may be created to represent the HOBs of a DSA.

```
hOBNB NAME BINDING
    SUBORDINATE OBJECT CLASS hOBMO AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS dSA AND SUBCLASSES;
    WITH ATTRIBUTE distinguishedName ;
    BEHAVIOUR
        hOBNBBehaviour BEHAVIOUR
            DEFINED AS ! Each HOB held by a DSA is named by the sequence of RDNs
                         forming the Distinguished Name of the root entry of the
                         subordinate naming context ! ;;
    CREATE WITH-REFERENCE-OBJECT ;
    DELETE ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-hOB-name-binding} ;
```

A.1.5.3 HOB package definition

The following definition specifies the package for the HOB managed object.

```
hOBPackage PACKAGE
    BEHAVIOUR hOBPackageBehaviour BEHAVIOUR
        DEFINED AS ! The information and actions permitted for managing HOBs ! ;;
    ATTRIBUTES
        distinguishedName GET,
        agreementID GET,
        agreementVersion GET,
        useDOP GET-REPLACE,
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalState GET,
        remoteAccessPoint GET-REPLACE,
        hOBRole GET ;
    NOTIFICATIONS
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":stateChange,
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalViolation
                hOBOperationalBindingID
                hOBDOPProblem,
```

"ITU-T Rec. X.723 (1993) | ISO/IEC 10165-5:1994":communicationsInformation
hOBOperationalBindingID;
REGISTERED AS {DirectoryManagementModule.id-mp-hOBPackage} ;

A.1.5.4 HOB notification parameters

The following parameter definitions are used with the notifications for HOBs:

hOBOperationalBindingID PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX DirectoryManagementModule.OperationalBindingID;
BEHAVIOUR hOBOperationalBindingIDBehaviour BEHAVIOUR
DEFINED AS The operational binding ID associated with the notification ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-hob-bind-id} ;

hOBDOPProblem PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX DirectoryManagementModule.OpBindingErrorParam ;
BEHAVIOUR hOBDOPProblemBehaviour BEHAVIOUR
DEFINED AS Reason why a DOP Error has been detected by the DSA ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-hob-dop-prob} ;

A.1.5.5 HOB notification behaviours

The following behaviour definitions are used with the notifications for HOBs:

hOBDOPErrorNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects a DOP error while maintaining the HOB.

The **serviceUser** field of the **operationViolation** notification contains the authenticated name of the DSA requesting the operation, or the peer DSA's AE-title.

The **serviceProvider** field of the **operationViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **operationViolation** notification contains any additional textual information to be conveyed in the notification.

The **operationalBindingID** parameter contains the operation binding ID of the HOB for which the error has been detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **DOPProblem** parameter contains an indication of the DOP error that was detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification. ! ;

hOBDOPCompleteNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA successfully completes a DOP operation.

The **serviceUser** field of the **communicationsInformation** notification contains the authenticated name of the DSA requesting the operation, or the peer DSA's AE-title.

The **serviceProvider** field of the **communicationsInformation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **communicationsInformation** notification contains any additional textual information to be conveyed in the notification.

The **operationalBindingID** parameter contains the operational binding ID of the HOB for which the operation has been completed and is conveyed as a parameter in the **additionalInformation** field of the **communicationsInformation** notification. ! ;

A.1.6 Shadowing Agreement managed object definitions

This subclause specifies the management definitions for a DSA's shadowing agreements.

A.1.6.1 Shadowing Agreement managed object class definition

The following definition specifies the managed objects used to represent a shadowing agreement held by a DSA.

```
shadowingAgreementMO MANAGED OBJECT CLASS

    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":top ;
    CHARACTERIZED BY shadowingAgreementPackage ;
    REGISTERED AS {DirectoryManagementModule.id-moc-shadowingAgreement} ;
```

A.1.6.2 Shadowing Agreement name binding definition

The following definition specifies the name binding for the Shadowing Agreement managed objects that may be created to represent the shadowing agreements of a DSA.

```
shadowingAgreementNB NAME BINDING

    SUBORDINATE OBJECT CLASS shadowingAgreementMO AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS dSA AND SUBCLASSES;
    WITH ATTRIBUTE distinguishedName ;
    BEHAVIOUR
        shadowingAgreementNBBehaviour BEHAVIOUR
            DEFINED AS ! Each shadowing agreement held by a DSA is named by the
                         sequence of RDNs forming the Distinguished Name of the root
                         entry of the naming context containing the unit of replication ! ;;
    CREATE WITH-REFERENCE-OBJECT ;
    DELETE ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-shadowingAgreement-nb} ;
```

A.1.6.3 Shadowing Agreement package definition

The following definition specifies the package for the HOB managed object.

```
shadowingAgreementPackage PACKAGE

    BEHAVIOUR shadowingAgreementPackageBehaviour BEHAVIOUR
        DEFINED AS ! The information and actions permitted for managing shadowing
                     agreements ! ;;
    ATTRIBUTES
        distinguishedName GET,
        agreementID GET,
        agreementVersion GET,
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalState GET,
        shadowingSubject GET-REPLACE,
        updateMode GET-REPLACE,
        masterAccessPoint GET,
        secondaryShadows GET-REPLACE ADD-REMOVE,
        useDOP GET-REPLACE,
        remoteAccessPoint GET-REPLACE,
        shadowingRole GET,
        lastUpdateTime GET-REPLACE,
        shadowingSchedule GET-REPLACE,
        nextUpdateTime GET-REPLACE ;
    ACTIONS
        updateShadow ;
    NOTIFICATIONS
        "CCITT Rec X.721 (1992) | ISO/IEC 10165-2:1992":stateChange,
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalViolation
                shadDOPProblem
                notificationAgreementID
                shadowProblem
                updateProblem
                notificationLastUpdateTime,
        "ITU-T Rec. X.723 (1993) | ISO/IEC 10165-5:1994":communicationsInformation
                notificationAgreementID ;
    REGISTERED AS {DirectoryManagementModule.id-mp-shadowingAgreementPackage} ;
```

A.1.6.4 Shadowing Agreement notification parameters

The following parameter definitions are used with the notifications for shadowing agreements.

shadDOPProblem PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX DirectoryManagementModule.OpBindingErrorParam ;
BEHAVIOUR shadDOPProblemBehaviour BEHAVIOUR
DEFINED AS ! Reason why a DOP Error has been detected by the DSA ! ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-shadowing-dop-prob} ;

shadowProblem PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX DirectoryManagementModule.ShadowProblem ;
BEHAVIOUR shadowProblemBehaviour BEHAVIOUR
DEFINED AS ! Reason why the shadow operation failed ! ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-shadowProblem} ;

updateProblem PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX DirectoryManagementModule.ShadowProblem ;
BEHAVIOUR updateProblemBehaviour BEHAVIOUR
DEFINED AS ! Reason why an updateError has been detected by the DSA ! ;;
REGISTERED AS {DirectoryManagementModule.id-mpa-updateProblem} ;

notificationsAgreementID PARAMETER
CONTEXT EVENT-INFO;
ATTRIBUTE agreementID ;
BEHAVIOUR notificationAgreementIDBehaviour BEHAVIOUR
DEFINED AS ! The agreement identification associated with the notification ! ;;
;

notificationLastUpdateTime PARAMETER
CONTEXT EVENT-INFO;
ATTRIBUTE lastUpdateTime ;
BEHAVIOUR notificationLastUpdateTimeBehaviour BEHAVIOUR
DEFINED AS ! The last update time associated with a shadowing agreement ! ;;
;

A.1.6.5 Shadowing Agreement notification behaviours

The following behaviour definitions are used with the notifications for shadowing agreements:

shadowUpdateCompleteNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA successfully completes a shadow update sequence.

The **serviceUser** field of the **communicationsInformation** notification contains the authenticated name of the DSA requesting the operation, or the peer DSA's AE-title.

The **serviceProvider** field of the **communicationsInformation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **communicationsInformation** notification contains any additional textual information to be conveyed in the notification.

The **notificationAgreementID** parameter contains the agreement ID of the **shadowingAgreement** for which the update has been completed and is conveyed as a parameter in the **additionalInformation** field of the **communicationsInformation** notification. ! ;

shadowErrorNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects a shadow error while performing a shadowing operation.

The **serviceUser** field of the **operationViolation** notification contains the authenticated name of the DSA requesting the operation, or the peer DSA's AE-title.

The **serviceProvider** field of the **operationViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **operationViolation** notification contains any additional textual information to be conveyed in the notification.

The **notificationAgreementID** parameter contains the agreement ID of the **shadowingAgreement** for which the error has been detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **shadowProblem** parameter contains an indication of the shadow error that was detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **notificationLastUpdateTime** parameter contains an indication of the time of last update for the DSA for the agreement identified in the **notificationAgreementID** parameter. The lastUpdateTime parameter is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification. ! ;

shadowDOPErrorNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA detects a DOP error while maintaining the shadowing agreement.

The **serviceUser** field of the **operationViolation** notification contains the authenticated name of the DSA requesting the operation, or the peer DSA's AE-title.

The **serviceProvider** field of the **operationViolation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **operationViolation** notification contains any additional textual information to be conveyed in the notification.

The **notificationAgreementID** parameter contains the agreement ID of the shadowing agreement for which the error has been detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification.

The **shadDOPProblem** parameter contains an indication of the DOP error that was detected and is conveyed as a parameter in the **additionalInformation** field of the **operationViolation** notification. ! ;

shadowDOPCompleteNotificationBehaviour BEHAVIOUR
DEFINED AS

! Notifications with this behaviour are generated whenever the DSA successfully completes a DOP operation.

The **serviceUser** field of the **communicationsInformation** notification contains the authenticated name of the DSA requesting the operation, or the peer DSA's AE-title.

The **serviceProvider** field of the **communicationsInformation** notification contains the AE-title of the DSA executing the request.

The **additionalText** field of the **communicationsInformation** notification contains any additional textual information to be conveyed in the notification.

The **notificationAgreementID** parameter contains the agreement ID of the shadowing agreement for which the operation has been completed and is conveyed as a parameter in the **additionalInformation** field of the **communicationsInformation** notification. ! ;

A.1.6.6 Shadowing Agreement Actions

The following actions are used for shadowing agreements:

updateShadow ACTION
BEHAVIOUR updateBehaviour BEHAVIOUR
DEFINED AS ! The action causes an out-of-band shadow update sequence to be initiated using the DISP protocol. !;;
REGISTERED AS {DirectoryManagementModule.id-mac-update};

A.2 Management of a Known DSA

The Known DSA is represented in the OSI Environment as an application process with application entities representing its communications capabilities. The Known DSA represents another DSA application entity with which the local Directory component interacts. This subclause identifies the managed objects used to represent and manage the Known DSA, its application entity invocations, application associations and operations.

A.2.1 Known DSA managed object definition

The following definition specifies the managed objects used to represent a Known DSA in an end system.

```
knownDSA MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.723 (1993) | ISO/IEC 10165-5:1994":communicationsEntity ;
    CHARACTERIZED BY knownDSAPackage ;
    REGISTERED AS {DirectoryManagementModule.id-moc-knownDSA} ;
```

A.2.2 Known DSA name binding definitions

The following definitions specify the naming relationship of Known DSAs to other managed objects. Known DSAs are named subordinate to DSA and DUA managed objects.

```
knownDSA-dSA NAME BINDING
    SUBORDINATE OBJECT CLASS knownDSA AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS dSA AND SUBCLASSES;
    WITH ATTRIBUTE dirCommonName ;
    CREATE WITH-REFERENCE-OBJECT ;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-knownDSA-dSA-name-binding} ;

knownDSA-dUA NAME BINDING
    SUBORDINATE OBJECT CLASS knownDSA AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS dUA AND SUBCLASSES;
    WITH ATTRIBUTE dirCommonName ;
    CREATE WITH-REFERENCE-OBJECT ;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-knownDSA-dUA-name-binding} ;
```

A.2.3 Known DSA package definition

The following definition specifies the package for Known DSAs.

```
knownDSAPackage PACKAGE
    BEHAVIOUR knownDSABehaviour BEHAVIOUR
    DEFINED AS ! This managed object class describes the information required to establish
                 an association to a neighbouring DSA, and contains association-related
                 statistics for the neighbour DSA. The CommunicationsAlarm notification is
                 sent when there is an abnormal termination of an association.!;;
    ATTRIBUTES
        dirCommonName GET,
        remoteAccessPoint GET-REPLACE,
        supportedApplicationContexts GET,
        credentials GET-REPLACE,
        reverseCredentials GET-REPLACE,
        dIRQOP GET-REPLACE,
        maxInboundAssocs GET-REPLACE,
        maxOutboundAssocs GET-REPLACE,
        timeOfLastAttempt GET,
        timeOfLastSuccess GET,
        currentActiveInboundAssocs GET,
        currentActiveOutboundAssocs GET,
        accumInboundAssocs GET,
        accumOutboundAssocs GET,
        accumFailedInboundAssocs GET,
        accumFailedOutboundAssocs GET,
        requestCounter GET,
        replyCounter GET,
        requestsFailedCounter GET ;
    NOTIFICATIONS
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":communicationsAlarm ;
    REGISTERED AS {DirectoryManagementModule.id-mp-knownDSAPackage} ;
```

A.3 Management of a Known DUA

The Known DUA is represented in the OSI Environment as an application process with application entities representing its communications capabilities. The Known DUA represents a DUA application entity with which the DSA interacts. This Section identifies the managed objects used to represent and manage the Known DUA, its application entity invocations, application associations and operations.

A.3.1 Known DUA managed object definition

The following definition specifies the managed objects used to represent a Known DUA in an end system.

```
knownDUA MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.723 (1993) | ISO/IEC 10165-5:1994":communicationsEntity ;
    CHARACTERIZED BY knownDUAPackage ;
    REGISTERED AS {DirectoryManagementModule.id-moc-knownDUA} ;
```

A.3.2 Known DUA name binding definition

The following definition specifies the naming relationship of Known DUAs to other managed objects. Known DUAs are named subordinate to a DSA.

```
knownDUA-dSA NAME BINDING
    SUBORDINATE OBJECT CLASS knownDUA AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS dSA AND SUBCLASSES;
    WITH ATTRIBUTE dirCommonName ;
    CREATE WITH-REFERENCE-OBJECT ;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-knownDUA-dSA-name-binding} ;
```

A.3.3 Known DUA package definition

The following definition specifies the package for Known DUAs.

```
knownDUAPackage PACKAGE
    BEHAVIOUR knownDUABehaviour BEHAVIOUR
        DEFINED AS ! This package contains the definitions that manage the view of DUAs as
                     viewed from the (local) DSA. The Communications Alarm notification is
                     sent when there is an abnormal termination of a DUA's association.!;;
    ATTRIBUTES
        dirCommonName GET,
        remoteAccessPoint GET,
        supportedApplicationContexts GET,
        credentials GET,
        reverseCredentials GET,
        timeOfLastAccess GET,
        currentActiveAssocs GET,
        accumAssocs GET ;
    NOTIFICATIONS
        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":communicationsAlarm;
    REGISTERED AS {DirectoryManagementModule.id-mp-knownDUAPackage} ;
```

A.4 Management of association

The Upper Layer Connection Endpoint represents an active association between the DSA and another DSA, or between the DSA and a DUA.

A.4.1 Upper Layer Connection Endpoint managed object definition

The following definition specifies the managed object used to represent an upper layer connection endpoint.

```
uLconnEnd MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.723 (1993) | ISO/IEC 10165-5:1994":singlePeerConnection ;
    CHARACTERIZED BY uLconnEndPackage ;
    REGISTERED AS {DirectoryManagementModule.id-moc-ULconnEnd} ;
```

A.4.2 Upper Layer Connection Endpoint name binding definitions

The following definitions specify the naming relationship of Upper Layer Connection Endpoints to other managed objects. Upper Layer Connection Endpoints are named subordinate to a Known DSAs and Known DUAs.

```
uLconnEnd-knownDSA NAME BINDING
    SUBORDINATE OBJECT CLASS uLconnEnd AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS knownDSA AND SUBCLASSES;
    WITH ATTRIBUTE associationId ;
    CREATE WITH-REFERENCE-OBJECT ;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-ULconnEnd-knownDSA} ;
```

```
uLconnEnd-knownDUA NAME BINDING
    SUBORDINATE OBJECT CLASS uLconnEnd AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS knownDUA AND SUBCLASSES;
    WITH ATTRIBUTE associationId ;
    CREATE WITH-REFERENCE-OBJECT ;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-ULconnEnd-knownDUA} ;
```

A.4.3 Upper Layer Connection Endpoint package definition

The following definition specifies the package for Upper Layer Connection Endpoints.

```
uLconnEndPackage PACKAGE
    BEHAVIOUR uLconnEndBehaviour BEHAVIOUR
    DEFINED AS ! This package defined the attributes for an application assocation ! ;;
    ATTRIBUTES
        callingAETitle GET,
        associationId GET,
        applicationContextInUse GET ;
    REGISTERED AS {DirectoryManagementModule.id-mp-ULconnEndPackage} ;
```

A.5 Management of a DUA

A DUA is represented in the OSI Environment as an application process with an application entity representing its communications capability. This Section identifies the managed objects used to represent and manage a DUA, its application entity invocations and application associations.

A.5.1 DUA managed object definition

The following definition specifies the DUA managed objects used to represent a DUA in an end system.

```
dUA MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":top ;
    CHARACTERIZED BY dUAPackage ;
    REGISTERED AS {DirectoryManagementModule.id-moc-dUA} ;
```

A.5.2 DUA package definition

The following definition specifies the package for DUAs.

```
dUAPackage PACKAGE
    BEHAVIOUR dUAPackageBehaviour BEHAVIOUR
        DEFINED AS ! This package contains the attributes and actions that manage the
                     view of the DUA as viewed from the DUA. No notifications are generated.
                     There are two actions to control which DSA the DUA should use. ! ;;
    ATTRIBUTES
        homeDSA GET-REPLACE SET-BY-CREATE,
        subSchema GET-REPLACE SET-BY-CREATE,
        dUATimeout GET-REPLACE SET-BY-CREATE ;
    ACTIONS
        useRemoteDSA, useHomeDSA;
    REGISTERED AS {DirectoryManagementModule.id-mp-dUAPackage} ;
```

A.5.3 DUA action definitions

The following definitions specify the actions for DUAs.

```
useRemoteDSA ACTION
    BEHAVIOUR useRemoteDSABehaviour BEHAVIOUR
        DEFINED AS ! Use one of the subordinate remotes Known instead of the home
                     DSA ! ;;
        REGISTERED AS {DirectoryManagementModule.id-mac-useRemoteDSA} ;

useHomeDSA ACTION
    BEHAVIOUR useHomeDSABehaviour BEHAVIOUR
        DEFINED AS ! Revert to using the home DSA ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mac-useHomeDSA} ;
```

A.6 Directory Service management

This subclause specifies the management definitions for a Directory Service.

A.6.1 Directory Service

A.6.1.1 Directory Service managed object definition

The following definition specifies the managed objects used to represent a Directory Service.

```
directoryService MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) ISO/IEC 10165-2:1992":top;
    CHARACTERIZED BY directoryServicePackage ;
    CONDITIONAL PACKAGES
        directoryInformationServicePackage PRESENT IF      'the DSA allows the Directory
                                                           service manager to control the
                                                           information processing capability
                                                           of the Directory',
        directoryControlServicePackage PRESENT IF 'the DSA allows the Directory service
                                                  manager to manage the operational
                                                  activity of the Directory' ;
    REGISTERED AS {DirectoryManagementModule.id-moc-disManagedObject} ;
```

A.6.1.2 Directory Service name binding definition

The following definition specifies the naming relationship of Directory Service managed objects to other managed objects. Directory Service managed objects are named subordinate to a Directory Customer.

```
directoryService-Customer NAME BINDING
    SUBORDINATE OBJECT CLASS directoryService AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS directoryCustomer AND SUBCLASSES;
    WITH ATTRIBUTE serviceIdentifier ;
    CREATE WITH-REFERENCE-OBJECT ;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-dis-Customer-name-binding} ;
```

A.6.1.3 Directory Service package definitions

The following definitions specify the package for the Directory Service managed object.

```
directoryServicePackage PACKAGE
    BEHAVIOUR directoryServicePackageBehaviour BEHAVIOUR
        DEFINED AS ! This package uses management definitions to be used for the
                     management of a Directory service ! ;;
        ATTRIBUTES
            serviceIdentifier GET-REPLACE SET-BY-CREATE,
            serviceDescription GET-REPLACE SET-BY-REPLACE ;
        REGISTERED AS {id-mp-dsPackage} ;

directoryInformationServicePackage PACKAGE
    BEHAVIOUR directoryInformationServiceBehaviour BEHAVIOUR
        DEFINED AS ! This package contains management definitions to be used for the
                     specification of a Directory information service. Certain attributes in
                     the package (including SizeLimit and TimeLimit) provide service
                     policy limits for use by the DSA. These limits overide any similar limits
                     that can be established for the DSA itself ! ;;
    ATTRIBUTES
        allowedDirectoryInformationServiceElement GET-REPLACE SET-BY-CREATE,
        disAllowedDirectoryInformationServiceElement GET-REPLACE SET-BY-CREATE,
        sizeLimit GET-REPLACE SET-BY-CREATE,
        timeLimit GET-REPLACE SET-BY-CREATE,
        accessor GET-REPLACE SET-BY-CREATE ;
    REGISTERED AS {DirectoryManagementModule.id-mp-disPackage} ;

directoryControlServicePackage PACKAGE
    BEHAVIOUR serviceControlServiceBehaviour BEHAVIOUR
        DEFINED AS ! This package contains management definitions to be used for the
                     operational control of a Directory service ! ;;
```

```
        ATTRIBUTES
            maxEntriesReturned GET-REPLACE SET-BY-CREATE,
            maxTimeForResults GET-REPLACE SET-BY-CREATE ;
        REGISTERED AS {DirectoryManagementModule.id-mp-dcsPackage} ;
```

A.6.2 Directory Customer

A.6.2.1 Directory Customer managed object definition

The following definition specifies the managed objects used to represent a Directory Customer.

```
directoryCustomer MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec.X.721 (1992) | ISO/IEC 10165-2:1992":top;
    CHARACTERIZED BY directoryCustomerPackage ;
    REGISTERED AS {DirectoryManagementModule.id-moc-dirCust} ;
```

A.6.2.2 Directory Customer name binding definition

The following definition specifies the name binding for the Directory Customer managed objects that may be created to represent the Directory Customers.

```
directoryCustomer-dMD NAME BINDING
    SUBORDINATE OBJECT CLASS directoryCustomer ;
    NAMED BY SUPERIOR OBJECT CLASS dMD ;
    WITH ATTRIBUTE directoryCustomerName ;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-DirCust-DMD} ;
```

A.6.2.3 Directory Customer package definition

The following defintion specifies the package for the Directory Customer managed object.

```
directoryCustomerPackage PACKAGE|
    BEHAVIOUR directoryCustomerBehaviour BEHAVIOUR
        DEFINED AS ! This package contains management defintions to be used for the
                     specification of Directory Customers ! ;;
    ATTRIBUTES
        directoryCustomerName GET-REPLACE SET-BY-CREATE,
        directoryCustomerAddress GET-REPLACE SET-BY-CREATE ;
    REGISTERED AS {DirectoryManagementModule.id-mp-dirCust};
```

A.6.3 Directory User

A.6.3.1 Directory User managed object

The following definition specifies the managed objects used to represent a Directory User.

```
directoryUser MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec.X.721 (1992) | ISO/IEC 10165-2:1992":top;
    CHARACTERIZED BY directoryUserPackage ;
    REGISTERED AS {DirectoryManagementModule.id-moc-dirUser};
```

A.6.3.2 Directory User name binding definition

The following definition specifies the name binding for the Directory User managed objects that may be created to represent the Directory Users.

```
directoryUser-directoryCustomer NAME BINDING
    SUBORDINATE OBJECT CLASS directoryUser ;
    NAMED BY SUPERIOR OBJECT CLASS directoryCustomer ;
    WITH ATTRIBUTE directoryUserName ;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS ;
    REGISTERED AS {DirectoryManagementModule.id-mnb-DirUser-DirCust} ;
```

A.6.3.3 Directory User package definition

The following defintion specifies the package for the Directory User managed object.

```
directoryUserPackage PACKAGE
    BEHAVIOUR directoryUserBehaviour BEHAVIOUR
        DEFINED AS ! This package conatins management definitions to be used for the
                     specification of Directory Users ! ;;
    ATTRIBUTES
        directoryUserName GET-REPLACE SET-BY-CREATE,
    REGISTERED AS {DirectoryManagementModule.id-mp-dirUser};
```

A.7 DMD

This subclause specifies the management definitions for a Directory Management Domain.

A.7.1 DMD managed object

The following definition specifies the managed objects used to represent a Directory Management Domain.

```
dMD MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec.X.721(1992) | ISO/IEC 10165-2:1992":top;
    CHARACTERIZED BY dMDPackage ;
    REGISTERED AS {DirectoryManagementModule.id-moc-dMD} ;
```

A.7.2 DMD package definition

The following defintion specifies the package for Directory Management Domains.

```
dMDPackage PACKAGE
    BEHAVIOUR dMDPackageBehaviour BEHAVIOUR
        DEFINED AS ! This package conatins management definitions to be used for the
                     specification of a Directory Management Domain ! ;;
    ATTRIBUTES
        dMDName GET-REPLACE SET-BY-CREATE ;
    NOTIFICATIONS;
    REGISTERED AS {DirectoryManagementModule.id-mp-dMD};
```

A.8 Definition of attributes

The following definitions specify the attributes for the Directory Managed Objects.

```
abandonOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR abandonsProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of abandon operations that
                     the DSA has processed. For each abandon operation that the DSA
                     processes, the DSA increases the counter by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-abandonOpsProc} ;

accessControlScheme ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtOID ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR accessControlSchemeBehaviour BEHAVIOUR
        DEFINED AS ! Defines which access control scheme is in operation in the
                     administrative area. This attribute maps to the accessControlScheme
                     Directory attribute ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-accessControlScheme} ;

accumAssocs ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR accumAssocsBehaviour BEHAVIOUR
        DEFINED AS ! This attribute defines the total accumulated associations from this
                     Network Element to a neighbouring Network Element ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-accumAssocs} ;

accumFailedInboundAssocs ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR accumFailedInboundAssocsBehaviour BEHAVIOUR
        DEFINED AS ! This attribute defines the total accumulated attempted inbound
                     associations from a neighbouring Network Element that failed! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-accumFailedInboundAssocs} ;
```

```
accumFailedOutboundAssocs ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR accumFailedOutboundAssocsBehaviour BEHAVIOUR
        DEFINED AS ! This attribute defines the total accumulated attempted outbound
                     associations to a neighbouring Network Element that failed! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-accumFailedOutboundAssocs} ;

accumInboundAssocs ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter ;
    BEHAVIOUR accumInboundAssocsBehaviour BEHAVIOUR
        DEFINED AS ! This attribute defines the total accumulated inbound associations from
                     this Network Element to a neighbouring Network Element ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-accumInboundAssocs} ;

accumOutboundAssocs ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter ;
    BEHAVIOUR accumOutboundAssocsBehaviour BEHAVIOUR
        DEFINED AS ! This attribute defines the total accumulated outbound associations from
                     this Network Element to a neighbouring Network Element ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-accumOutboundAssocs} ;

accessor ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.Accessors ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR accessorBehaviour BEHAVIOUR
        DEFINED AS ! The identifier for a particular Directory information service
                     accessor! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-accessor} ;

accessPoint ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.AccessPoint ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR accessPointBehaviour BEHAVIOUR
        DEFINED AS ! The myAccessPoint attribute of the root DSE in the DSA. This
                     attribute contains the presentation address, protocol information
                     and AE Title of the DSA ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-accessPoint} ;

addEntryOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR addsProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of addEntry operations that
                     the DSA has processed in the evaluation phase. For each addEntry
                     operation that the DSA evaluates, the DSA increases the counter
                     by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-addEntryOpsProc} ;

administrativeRole ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.AdministrativeRole ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR administrativeRoleBehaviour BEHAVIOUR
        DEFINED AS ! Identifies the start an administrative areas This attribute maps to
                     the administrativeRole Directory attribute. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-administrativeRole} ;

adminLimitExceeded ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR adminLimitExceededBehaviour BEHAVIOUR
        DEFINED AS ! The number of administrative limit exceeded errors reported by the
                     DSA! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-adminLimitExceeded} ;

affectsMultipleDSAs ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR affectsMultipleDSAsBehaviour BEHAVIOUR
        DEFINED AS ! The number of affects Multiple DSA errors reported by the DSA ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-affectsMultipleDSAs} ;

aliasedEntryName ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.DistinguishedName ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR aliasedEntryNameBehaviour BEHAVIOUR
```

```
        DEFINED AS ! Holds the alias target. This attribute maps to the aliasedEntryName
                Directory. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-aliasedEntryName} ;

agreementID ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryShadowAbstractService.MgtInteger ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR agreementIDBehaviour BEHAVIOUR
        DEFINED AS ! The agreement identification for an operational binding agreement ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-agreementID} ;

agreementVersion ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtInteger ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR agreementVersionBehaviour BEHAVIOUR
        DEFINED AS ! The agreement version for an operational binding agreement ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-agreementVersion} ;

aliasDereferences ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR aliaseDereferencesBehaviour BEHAVIOUR
        DEFINED AS ! The number of alias dereferences performed by the DSA ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-aliasDereferences} ;

aliasDereferencingProblem ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR aliasDereferencingProblemBehaviour BEHAVIOUR
        DEFINED AS ! The number of alias dereferencing problem errors reported by the
                DSA ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-aliasDereferencingProblem} ;

aliasProblem ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR aliasProblemBehaviour BEHAVIOUR
        DEFINED AS ! The number of alias problem errors reported by the DSA ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-aliasProblem} ;

allowedDirectoryInformationServiceElement ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.DirectoryInformationServiceElement ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR allowedDirectoryInformationServiceElementBehaviour BEHAVIOUR
        DEFINED AS ! The permitted Directory information service elements ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-allowedInfoService} ;

applicationContextInUse ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.ApplicationContext ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR applicationContextInUseBehaviour BEHAVIOUR
        DEFINED AS ! The application context in use on an association ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-applicationContextInUse} ;

associationId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.AssociationId ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR associationIdBehaviour BEHAVIOUR
        DEFINED AS ! The association Identifier for the application association ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-associationId} ;

attributeTypes ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.AttributeTypeDescription ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR attributeTypesBehaviour BEHAVIOUR
        DEFINED AS ! Lists the attribute types for use in the subschema administrative
                area. This attribute maps to the attributeTyped Directory Attribute. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-attributeTypes} ;

callingAETitle ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtName ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR callingAETitleBehaviour BEHAVIOUR
        DEFINED AS ! The AE Title of the calling entity ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-callingAETitle} ;
```

```
chainedAbandonOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainedAbandonsProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chained abandon
                     operations that the DSA has processed. For each chained abandon
                     operation that the DSA processes, the DSA increases the counter
                     by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chAbandonOpsProc} ;

chainedAddEntryOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainedAddsProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chainedAddEntry
                     operations that the DSA has processed in the evaluation phase. For each
                     chainedAddEntry operation that the DSA evaluates, the DSA increases
                     the counter by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chAddEntryOpsProc} ;

chainedCompareOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainedComparesProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chainedCompare operations
                     that the DSA has processed in the evaluation phase. For each chained
                     compare operation that the DSA evaluates, the DSA increases the
                     counter by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chCompareOpsProc} ;

chainedListOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainedListsProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chainedList operations that
                     the DSA has processed. For each chainedList operation that the DSA
                     processes, the DSA increases the counter by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chListOpsProc} ;

chainedModifyEntryOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainedModifiesProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chainedModifyEntry
                     operations that the DSA has processed. For each chainedModifyEntry
                     operation that the DSA processes, the DSA increases the counter
                     by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chModifyEntryOpsProc} ;

chainedModifyDNOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainedModifyDNsProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chainedModifyDN
                     operations that the DSA has processed. For each chainedModifyDN
                     operation that the DSA processes, the DSA increases the counter
                     by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chModifyDNOpsProc} ;

chainedReadOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainedReadsProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chainedRead operations
                     that the DSA has processed in the evaluation phase. For each
                     chainedRead operation that the DSA evaluates, the DSA increases
                     the counter by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chReadOpsProc} ;

chainedRemoveEntryOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainedRemovesProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chainedRemoveEntry
                     operations that the DSA has processed in the evaluation phase. For each
                     chainedRemoveEntry operation that the DSA evaluates, the DSA
                     increases the counter by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chRemoveEntryOpsProc} ;
```

```
chainedSearch1LevelOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chained1LevelSearchesProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chainedSearch operations
                     that the DSA has processed that refer to the base object's immediate
                     subordinates. For each such operation that the DSA processes, the DSA
                     increases the counter by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chSearch1LevelOpsProc} ;

chainedSearchBaseOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainedBaseSearchesProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chainedSearch operations
                     that the DSA has processed that only refer to the base object. For each
                     operation that the DSA processes, the DSA increases the counter
                     by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chSearchBaseOpsProc} ;

chainedSearchSubtreeOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainedSubtreeSearchesProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of chained search operations
                     that the DSA has processed that refer to a whole subtree. For each such
                     operation that the DSA processes, the DSA increases the counter
                     by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chSearchSubtreeOpsProc} ;

chainings ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR chainingsBehaviour BEHAVIOUR
        DEFINED AS ! The number of chained operations initiated by the DSA ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-chainings} ;

collectiveExclusions ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtOID ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR collectiveExclusionsBehaviour BEHAVIOUR
        DEFINED AS ! The list of collective attributes which are excluded
                     from the corresponding Directory entry. This attribute maps to the
                     collectiveExclusions Directory attribute ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-collectiveExclusions} ;

compareOperationsProcessed ATTRIBUTE
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":counter-Threshold ;
    BEHAVIOUR comparesProcessedBehaviour BEHAVIOUR
        DEFINED AS ! This attribute is used to count the number of compare operations that
                     the DSA has processed in the evaluation phase. For each compare
                     operation that the DSA evaluates, the DSA increases the counter
                     by 1. ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-compareOpsProc} ;

consumerKnowledge ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.ConsumerInformation ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR consumerKnowledgeBehaviour BEHAVIOUR
        DEFINED AS ! Holds the knowledge about the consumer of shadow information
                     supplied by this DSA. This attribute maps to the consumerKnowledge
                     Directory attribute ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-consumerKnowledge} ;

copyEntries ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtInteger;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR copyEntriestBehaviour BEHAVIOUR
        DEFINED AS ! The number of entry copies held by the DSA ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-copyEntries} ;

createTimestamp ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtGeneralizedTime ;
    MATCHES FOR EQUALITY, ORDERING ;
    BEHAVIOUR createTimestampBehaviour BEHAVIOUR
```

```
            DEFINED AS ! Holds the time at which this DSE was created. This attribute maps
                    to the createTimestamp Directory attribute. ! ;;
        REGISTERED AS {DirectoryManagementModule.id-mat-createTimestamp} ;

    creatorsName ATTRIBUTE
        WITH ATTRIBUTE SYNTAX DirectoryManagementModule.DistinguishedName ;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR creatorsNameBehaviour BEHAVIOUR
            DEFINED AS ! Holds the distinguished name of the creator of
                    the DSE. The attribute maps to the creatorsName Directory attribute. ! ;;
        REGISTERED AS {DirectoryManagementModule.id-mat-creatorsName} ;

    credentials ATTRIBUTE
        WITH ATTRIBUTE SYNTAX DirectoryManagementModule.Credentials ;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR credentialsBehaviour BEHAVIOUR
            DEFINED AS ! This attribute contains the credentials sent with a bind request !;;
        REGISTERED AS {DirectoryManagementModule.id-mat-credentials} ;

    currentActiveAssocs ATTRIBUTE
        DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":gauge-Threshold ;
        BEHAVIOUR currentActiveAssocsBehaviour BEHAVIOUR
            DEFINED AS ! This attribute defines the current total of active associations from this
                    Network Element to a neighbouring Network Element ! ;;
        REGISTERED AS {DirectoryManagementModule.id-mat-currentActiveAssocs} ;

    currentActiveInboundAssocs ATTRIBUTE
        DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":gauge-Threshold ;
        BEHAVIOUR currentActiveInboundAssocsBehaviour BEHAVIOUR
            DEFINED AS ! This attribute defines the current total of active inbound associations
                    from this Network Element to neighbouring Network Element ! ;;
        REGISTERED AS {DirectoryManagementModule.id-mat-currentActiveInboundAssocs} ;

    currentActiveOutboundAssocs ATTRIBUTE
        DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":gauge-Threshold ;
        BEHAVIOUR currentActiveOutboundAssocsBehaviour BEHAVIOUR
            DEFINED AS ! This attribute defines the current total of active outbound associations
                    from this Network Element to a neighbouring Network Element ! ;;
        REGISTERED AS {DirectoryManagementModule.id-mat-currentActiveOutboundAssocs} ;

    dAPAssociationTimeout ATTRIBUTE
        WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtInteger ;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR dAPAssociationTimeoutBehaviour BEHAVIOUR
            DEFINED AS ! The number of seconds after which the DSA shall
                    timeout a quiescent DAP association ! ;;
        REGISTERED AS {DirectoryManagementModule.id-mat-dAPAssociationsTimeout} ;

    dIRQOP ATTRIBUTE
        WITH ATTRIBUTE SYNTAX EnhancedSecurity.DIRQOP ;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR dISPDIRQOPBehaviour BEHAVIOUR
            DEFINED AS ! The Directory quality of service used by a Directory component when
                    it communicates with a peer Directory component ! ;;
        REGISTERED AS {DirectoryManagementModule.id-mat-dIRQOP} ;

    dISPAssociationEstablishment ATTRIBUTE
        WITH ATTRIBUTE SYNTAX DirectoryManagementModule.AssociationEstablishment ;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR dISPAssociationEstablishmentBehaviour BEHAVIOUR
            DEFINED AS ! The types of association establishment supported by
                    the DSA for DISP association ! ;;
        REGISTERED AS {DirectoryManagementModule.id-mat-dISPAssociationEstablishment} ;

    dISPAssociationTimeout ATTRIBUTE
        WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtInteger ;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR dISPAssociationTimeoutBehaviour BEHAVIOUR
            DEFINED AS ! The number of seconds after which the DSA shall
                    timeout a quiescent DISP association ! ;;
        REGISTERED AS {DirectoryManagementModule.id-mat-dISPAssociationTimeout} ;
```

```
dOPAssociationEstablishment ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.AssociationEstablishment ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR dOPAssociationEstablishmentBehaviour BEHAVIOUR
        DEFINED AS ! The types of association establishment supported by
                     the DSA for DOP association ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-dOPAssociationEstablishment} ;

dOPAssociationTimeout ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtInteger ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR dOPAssociationTimeoutBehaviour BEHAVIOUR
        DEFINED AS ! The number of seconds after which the DSA shall
                     timeout a quiescent DOP association ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-dOPAssociationTimeout} ;

dSAActiveAssociationsThreshold ATTRIBUTE
    DERIVED FROM "ITU-T Rec. X.721 (1992) | ISO/IEC 10165-2:1992":gauge-Threshold ;
    BEHAVIOUR dSAActiveAssociationThresholdBehaviour BEHAVIOUR
        DEFINED AS ! This value is an indication of the total number of the DSA's
                     active associations. The crossing of a high-value threshold will
                     cause the generation of the notification
                     "dSAActiveAssociationsNotification"! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-dSAActiveAssociations} ;

dSAScopeOfChaining ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.DSAScopeOfChainingValue ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR dSAScopeOfChainingBehaviour BEHAVIOUR
        DEFINED AS ! The limitation on the DSA of chaining to one of DMD, country or
                     global scope ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-dSAScopeOfChaining} ;

dSAScopeOfReferral ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.DSAScopeOfReferralValue ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR dSAScopeOfReferralBehaviour BEHAVIOUR
        DEFINED AS ! The limitation on the DSA of referral to one of DMD, country or
                     global scope ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-dSAScopeOfReferral} ;

dSPAssociationEstablishment ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.AssociationEstablishment ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR dSPAssociationEstablishmentBehaviour BEHAVIOUR
        DEFINED AS ! The types of association establishment supported by
                     the DSA for DSP association ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-dSPAssociationEstablishment} ;

dSPAssociationTimeout ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtInteger ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR dSPAssociationTimeoutBehaviour BEHAVIOUR
        DEFINED AS ! The number of seconds after which the DSA shall
                     timeout a quiescent DSP association ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-dSPAssociationTimeout} ;

    dUATimeout ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtInteger ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR dUATimeoutBehaviour BEHAVIOUR
        DEFINED AS ! The number of seconds of inactivity on the association before the
                     association is aborted ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-dUATimeout} ;

dirCommonName ATTRIBUTE
    WITH ATTRIBUTE SYNTAX DirectoryManagementModule.MgtCommonName ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR dirCommonNameBehaviour BEHAVIOUR
        DEFINED AS ! Holds the name of the Directory component ! ;;
    REGISTERED AS {DirectoryManagementModule.id-mat-dirCommonName} ;
```