INTERNATIONAL STANDARD

ISO/IEC 9594-9

Seventh edition
2017-05

Information technology — Open Systems Interconnection — The Directory —

Part 9: Replication

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — L'annuaire —

Partie 9: Duplication

Reference number
ISO/IEC 9594-9:2017(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This seventh edition cancels and replaces the sixth edition (ISO/IEC 9594-9:2014), which has been technically revised.

This document was prepared by ISO/IEC JTC 1, *Information technology*, SC 6, *Telecommunications and information exchange between systems*, in collaboration with ITU-T. The identical text is published as ITU-T X.525 (10/2016).

A list of all parts in the ISO/IEC 9594 series, published under the general title *Information technology — Open Systems Interconnection — The Directory*, can be found on the ISO website.

CONTENTS

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 that 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.

This Recommendation | International Standard defines the replication capabilities provided by Directory system agents (DSAs) to improve the level of service to Directory users.

This Recommendation | International Standard provides the foundation frameworks upon which industry profiles can be defined by other standards groups and industry forums. Many of the features defined as optional in these frameworks may be mandated for use in certain environments through profiles. This eighth edition technically revises and enhances the seventh edition of this Recommendation | International Standard.

This eighth edition specifies versions 1 and 2 of the Directory protocols.

The first and second editions specified only version 1. Most of the services and protocols specified in this edition are designed to function under version 1. However, some enhanced services and protocols, e.g., signed errors, will not function unless all Directory entities involved in the operation have negotiated version 2. Whichever version has been negotiated, differences between the services and between the protocols defined in the eight editions, except for those specifically assigned to version 2, are accommodated using the rules of extensibility defined in Rec. ITU-T X.519 | ISO/IEC 9594-5.

Annex A, which is an integral part of this Recommendation | International Standard, provides the ASN.1 module for the Directory shadow abstract service.

Annex B, which is not an integral part of this Recommendation | International Standard, lists the amendments and defect reports that have been incorporated to form this edition of this Recommendation | International Standard.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – Open Systems Interconnection – The Directory: Replication

1 Scope

This Recommendation | International Standard specifies a shadow service which Directory system agents (DSAs) may use to replicate Directory information. The service allows Directory information to be replicated among DSAs to improve service to Directory users. The shadowed information is updated, using the defined protocol, thereby improving the service provided to users of the Directory.

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 Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

– Recommendation ITU-T X.500 (2016) | ISO/IEC 9594-1:2017, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services.*

– Recommendation ITU-T X.501 (2016) | ISO/IEC 9594-2:2017, *Information technology – Open Systems Interconnection – The Directory: Models.*

– Recommendation ITU-T X.509 (2016) | ISO/IEC 9594-8:2017, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks.*

– Recommendation ITU-T X.511 (2016) | ISO/IEC 9594-3:2017, *Information technology – Open Systems Interconnection – The Directory: Abstract service definition.*

– Recommendation ITU-T X.518 (2016) | ISO/IEC 9594-4:2017, *Information technology – Open Systems Interconnection – The Directory: Procedures for distributed operation.*

– Recommendation ITU-T X.519 (2016) | ISO/IEC 9594-5:2017, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications.*

– Recommendation ITU-T X.520 (2016) | ISO/IEC 9594-6:2017, *Information technology – Open Systems Interconnection – The Directory: Selected attribute types.*

– Recommendation ITU-T X.521 (2016) | ISO/IEC 9594-7:2017, *Information technology – Open Systems Interconnection – The Directory: Selected object classes.*

– Recommendation ITU-T X.680 (2015) | ISO/IEC 8824-1:2015, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*

3 Definitions

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

3.1 Basic Directory definitions

The following term is defined in Rec. ITU-T X.500 | ISO/IEC 9594-1:

– *(the) Directory*.

3.2 Directory model definitions

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

a) distinguished name;
b) Directory information tree (DIT);
c) DSA-specific entry (DSE);
d) DSA information model;
e) DSA information tree;
f) Directory system agent (DSA).

3.3 Abstract service definitions

The following term is defined in Rec. ITU-T X.511 | ISO/IEC 9594-3:

a) request;
b) requestor.

3.4 Distributed operation definitions

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

a) access point;
b) knowledge information;
c) name resolution;
d) naming context;
e) non-specific subordinate reference;
f) subordinate reference.

3.5 Protocol definitions

The following term is defined in Rec. ITU-T X.519 | ISO/IEC 9594-5:

a) application-association.

3.6 Replication definitions

The following terms are defined in this Recommendation | International Standard:

3.6.1 area prefix: The sequence of RDNs and associated administrative information common to all entries within a replicated area.

3.6.2 attribute completeness: Indicates whether or not all user attributes are included in an entry-copy.

3.6.3 cache-copy: A copy of an entry (or part of an entry) whose consistency with its corresponding entry is maintained by means outside the scope of this Directory Specification.

3.6.4 caching: The process of creating cache copies. This process is outside the scope of this Directory Specification.

3.6.5 consumer reference: The access point of the shadow consumer.

3.6.6 entry-copy: Shadowed information from an entry.

3.6.7 extended knowledge: Those subordinate and non-specific subordinate references that would be included as subordinate knowledge if the replicated area were extended to the lower boundary of the naming context.

3.6.8 master DSA: The DSA which has administrative authority for a naming context. All adds, deletes and modifications to entries in this naming context are done by the master DSA. The master DSA may enter into shadowing agreements with other DSAs to provide copies of a subset of a naming context (see unit of replication).

3.6.9 primary shadowing: Shadowing where the shadow supplier is the master DSA.

3.6.10 replicated area: A subtree of the DIT for purposes of shadowing.

3.6.11 replication: The process by which copies of entry and operational information are held by DSAs other than the master DSA.

3.6.12 replication base entry: The distinguished name of the root vertex of a replicated area.

3.6.13 secondary shadowing: Shadowing where the shadow supplier is not the master DSA.

3.6.14 shadow consumer: A DSA that receives shadowed information.

3.6.15 shadow operational binding: The relationship between two DSAs, one acting as a supplier of replicated information and the other as its consumer.

3.6.16 shadow service: The service provided to perform shadowing between two DSAs that have entered into one or more shadowing agreements.

3.6.17 shadow supplier: A DSA that provides shadowed information. This DSA may or may not be the master DSA.

3.6.18 shadowed DSA specific entry (SDSE): A unit of shadowed information which is associated with a specific name; it represents the information taken from a DSE which is shadowed.

3.6.19 shadowed information: The complete set of information associated with a unit of replication. Shadowed information is conceptually held both by the shadow supplier and the shadow consumer for the purposes of the shadow protocol and comprises a tree shaped structure of shadowed DSEs.

3.6.20 shadowing: Replication between two DSAs whereby shadowed information is copied and maintained using the Directory Information Shadowing Protocol.

3.6.21 shadowing agreement: The terms specific to a particular agreement required for shadowing to occur between a pair of DSAs.

3.6.22 subordinate completeness: Indicates whether or not subordinate knowledge is complete for an entry-copy.

3.6.23 supplier reference: The access point of the shadow supplier.

3.6.24 unit of replication: A specification of the information to be shadowed, including (optionally) subordinate knowledge information.

4 Abbreviations

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

ACI	Access Control Information
DIB	Directory Information Base
DISP	Directory Information Shadowing Protocol
DIT	Directory Information Tree
DMD	Directory Management Domain
DSA	Directory System Agent
DSE	DSA-Specific Entry
DUA	Directory User Agent
LDAP	Lightweight Directory Access Protocol
RDN	Relative Distinguished Name
SDSE	Shadowed DSA-Specific Entry
TCP/IP	Transmission Control Protocol/Internet Protocol

5 Conventions

The term "Directory Specification" (as in "this Directory Specification") shall be taken to mean Rec. ITU-T X.525 | ISO/IEC 9594-9. The term "Directory Specifications" shall be taken to mean the X.500-series Recommendations and all parts of ISO/IEC 9594.

This Directory Specification uses the term first edition systems to refer to systems conforming to the first edition of the Directory Specifications, i.e., the 1988 edition of the CCITT X.500-series Recommendations and the ISO/IEC 9594:1990 edition.

This Directory Specification uses the term second edition systems to refer to systems conforming to the second edition of the Directory Specifications, i.e., the 1993 edition of the ITU-T X.500-series Recommendations and the ISO/IEC 9594:1995 edition.

This Directory Specification uses the term third edition systems to refer to systems conforming to the third edition of the Directory Specifications, i.e., the 1997 edition of the ITU-T X.500-series Recommendations and the ISO/IEC 9594:1998 edition.

This Directory Specification uses the term fourth edition systems to refer to systems conforming to the fourth edition of the Directory Specifications, i.e., the 2001 editions of Recs ITU-T X.500, ITU-T X.501, ITU-T X.511, ITU-T X.518, ITU-T X.519, ITU-T X.520, ITU-T X.521, ITU-T X.525, and ITU-T X.530, the 2000 edition of ITU-T X.509, and parts 1-10 of the ISO/IEC 9594:2001 edition.

This Directory Specification uses the term fifth edition systems to refer to systems conforming to the fifth edition of the Directory Specifications, i.e., the 2005 edition of the ITU-T X.500-series Recommendations and the ISO/IEC 9594:2005 edition.

This Directory Specification uses the term sixth edition systems to refer to systems conforming to the sixth edition of the Directory Specifications, i.e., the 2008 edition of the ITU-T X.500-series Recommendations and the ISO/IEC 9594:2008 edition.

This Directory Specification uses the term seventh edition systems to refer to systems conforming to the seventh edition of the Directory Specifications, i.e., the 2012 edition of the ITU-T X.500-series Recommendations and the ISO/IEC 9594:2014 edition.

This Directory Specification uses the term eighth edition systems to refer to systems conforming to the eighth edition of the Directory Specifications, i.e., the 2016 edition of the ITU-T X.500-series Recommendations and the ISO/IEC 9594:2017 edition.

This Directory Specification presents ASN.1 notation in the bold Courier New typeface. When ASN.1 types and values are referenced in normal text, they are differentiated from normal text by presenting them in the bold Courier New typeface. The names of procedures, typically referenced when specifying the semantics of processing, are differentiated from normal text by displaying them in bold Times New Roman. Access control permissions are presented in italicized Times New Roman.

If the items in a list are numbered (as opposed to using "-" or letters), then the items shall be considered steps in a procedure.

6 Replication in the Directory

Replicated (copied) information can exist in the Directory. Shadowing is the mechanism for replication defined in this Directory Specification. Directory information can also be replicated by means outside this Directory Specification, such as caching. Any such alternative means of replication will need to ensure that exactly one instance of each replicated entry is identified as the master copy if the Directory and DSA abstract services are to be used.

Service controls provide the ability to control whether replicated information may be used in support of Directory operations, regardless of the replication mechanism used to acquire the copy. DISP is protected by the underlying protocol as defined in Rec. ITU-T X.519 | ISO/IEC 9594-5.

6.1 Caching

One method of replicating Directory information is caching. Caching procedures are considered to be almost entirely governed by local policies, and therefore outside the scope of this Directory Specification.

6.2 Shadowing

Another method of replicating Directory information is shadowing. An overview of the Directory information shadow service is found in clause 7. Before shadowing can occur, an agreement, covering the conditions under which shadowing may occur is required. Although such agreements may be established in a variety of ways, such as policy statements covering all DSAs within a given Directory management domain (DMD), the shadowing is always between a pair of DSA. The technical parameters for the subsequent shadowing are specified as part of the resulting shadowing agreement. Components of the shadowing agreement are defined in clause 9.

Once the terms of the agreement have been established, the DSAs may initiate, modify and subsequently terminate the shadowing agreement. This may be done through a shadow operational binding as defined in clause 8.

This shadowing service for the Directory is based on the models established in Rec. ITU-T X.501 | ISO/IEC 9594-2, to satisfy the requirements outlined in Rec. ITU-T X.500 | ISO/IEC 9594-1. The protocol specification for shadowing and conformance requirements are provided in Rec. ITU-T X.519 | ISO/IEC 9594-5. In addition, this Directory Specification provides the definition of an operational binding for the purpose of initiating, modifying, and terminating shadowing agreements between DSAs. This operational binding type is defined using the tools specified in Rec. ITU-T X.501 | ISO/IEC 9594-2.

The Directory information shadow service is defined in clause 10. The actual shadowing occurs through the set of operations defined in clause 11. These operations accommodate the transfer of Directory information and updates to the shadowed information.

The use of shadowed information by a DSA to satisfy a Directory request is described in Rec. ITU-T X.518 | ISO/IEC 9594-4.

6.3 Shadowing functional model

In the standardized form of Directory replication, termed *shadowing*, a DSA may assume the role of *shadow supplier*, the source of shadowed information, or *shadow consumer*, the recipient of shadowed information. The role played by a DSA when engaging in standardized replication activities (shadow supplier or shadow consumer) is always with respect to another DSA which plays the reciprocal role (shadow consumer or shadow supplier).

A given DSA may assume both roles, either:

- with respect to different DSAs for the same or different units of replication; or
- with respect to a single DSA (which plays the reciprocal role) for different units of replication.

The shadowing functional model addresses two approaches to shadowing Directory information:

- a *primary shadowing* policy requires that each shadow consumer receives its updates directly from the master DSA for the unit of replication;
- a *secondary shadowing* policy permits a shadow consumer to assume the shadow supplier role with respect to shadow consumers not having a shadowing agreement directly with the master DSA.

The characteristics of these two policies and their approach to addressing performance, availability, reliability and recovery are described below.

6.3.1 Primary shadowing

Figure 1 depicts primary shadowing. In this case, the shadowing policy in effect has the following characteristics:

a) the master DSA is the only shadow supplier for a replicated area;

b) each shadow consumer has a direct shadowing agreement with the master DSA;

c) only read, compare, search, and list operations may be performed at a shadow consumer holding shadowed information. All modification operations are directed to the master DSA.

Because it allows for the placement of copies of often requested information, or knowledge of it, closer to the requestor, this approach may be used to satisfy the performance requirement. Also, because this approach provides for the redundancy of individual entry or knowledge information, it is possible, in a primitive sense, to provide for availability, reliability, and recovery.

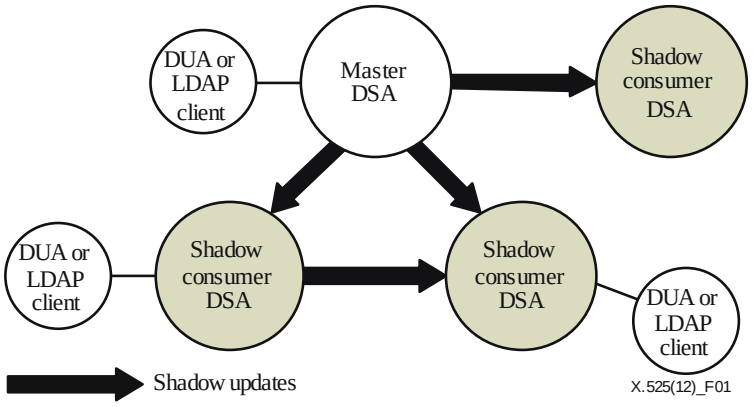

Figure 1 – Primary shadowing

6.3.2 Secondary shadowing

Figure 2 depicts secondary shadowing. In this case, the shadowing policy in effect has the following characteristics:

a) The master DSA is not the only shadow supplier for a replicated area. Only some shadow consumers have a direct shadowing agreement with the master DSA as their shadow supplier.

b) Other shadow consumers may have a shadowing agreement with a shadow supplier that is not the master for the unit of replication. The shadowing agreements between the master DSA and its direct shadow consumers may, however, have an impact on secondary shadowing agreements.

c) Only read, compare, search, and list operations may be performed at a shadow consumer holding shadowed information. All modification operations are directed to the master DSA, either directly (if a secondary shadow consumer DSA has knowledge of the master DSA) or indirectly via the shadow supplier DSA(s).

Secondary shadowing is very similar to primary shadowing in the way that it provides for performance, availability, reliability and recovery. It differs in that it relieves the single master DSA of the burden of directly supplying all shadow consumers with the shadowed information. This is a desirable combination in environments where a large number of shadow consumers are holding the same shadowed information.

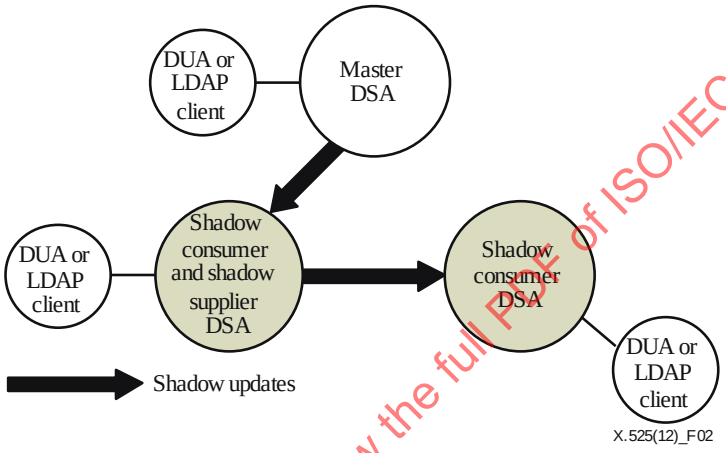

Figure 2 – Secondary shadowing

7 Shadowing in the Directory

The Directory information shadow service defined here provides the Directory with a standardized mechanism to provide and support shadowed information. In outline, the shadow supplier maintains, for each shadowing agreement, information which is to be shadowed (the shadowed information). This information is replicated by protocol exchange between the shadow supplier and the shadow consumer. The information to be shadowed is all or a subset of the information held by the shadow supplier's DSA information tree. The shadow consumer's shadowed information becomes part of its DSA information tree.

To use the Directory information shadow service, the administrative authorities of two DSAs must first reach an agreement on the terms under which shadowing will take place. This agreement, and the technical specification related to this agreement (the shadowing agreement), is discussed in 7.1. A description of the manner in which shadowed information is represented for the purposes of shadowing is provided in 7.2. The actual transfer of this shadowed information from the shadow supplier to the shadow consumer is accomplished by means of a set of shadow operations, which are introduced in 7.3.

The use of shadowed information to satisfy Directory requests is described in Rec. ITU-T X.518 | ISO/IEC 9594-4.

7.1 Shadowing agreement

Before shadowing can occur, an agreement for shadowing is established between the administrative authorities of the Directory management domains involved in the shadowing. This agreement for shadowing may be multilateral with respect to DSAs, in that it may cover all shadowing permitted among the set of DSAs concerned. The agreement may include any set of terms acceptable to the administrative authorities. For example, the agreement may specify policy information related to security, charging, or other special conditions.

A shadowing agreement is the specific agreement for a particular instance of shadowing between a pair of DSAs (the shadow consumer DSA and the shadow supplier DSA). This agreement may be explicit (e.g., contractual) or implicit

(e.g., covered by the general terms of an agreement for shadowing as defined above). Each shadowing agreement has a unique identifier used in all protocol exchanges associated with the agreement. Other parameters of a shadowing agreement include a specification of the unit of replication, the update mode and possibly the access point of the master DSA for the shadowed information. Access control information is always included in shadowed information and therefore need not be explicitly specified.

Initially, the representation of the shadowing agreement within a DSA (shadow supplier or shadow consumer) is created by an off-line administrative process. It represents essentially a template whose technical parameter values are subsequently validated during the initiating phase of the agreement and possibly modified during modification operations on the agreement. The method of storing this agreement is beyond the scope of this Directory Specification. Some technical aspects of the shadowing agreement may be exchanged via protocol and are discussed in detail in clause 9.

Although the shadowing agreement will normally provide a true representation of the technical parameters related to the Directory information shadow service, there may be exceptional cases in which policy overrides the technical specification resulting in a service inconsistency. For example, there may be certain attributes or attribute values that are withheld for security reasons. It may be the case that security policy prevents disclosing the mere existence of these attributes, in which case it would be a violation to represent in the shadowing agreement the fact that they are being withheld. In this type of situation, the behaviour of the shadow supplier DSA will be as if the technical specification were a true representation. Thus, users with access to the sensitive data will receive different views of the affected entries, depending on whether they access the master or a shadow consumer.

7.2 Shadowed information

Shadowed information is the logical set of information which is replicated by the shadow consumer. A replicated area is a subtree of the DIT defined for purposes of shadowing. The three components of shadowed information are:

a) *Prefix information*: Information relevant to entries within the replicated area which, with respect to the DSA information model, is positioned between the area prefix and the root DSA-specific entry (DSE). This may contain administrative entry and subentry information.

b) *Area information*: Information about DSEs whose names fall within the replicated area.

c) *Subordinate information*: Information about knowledge references subordinate to the replicated area.

Figure 3 illustrates the derivation of shadowed information.

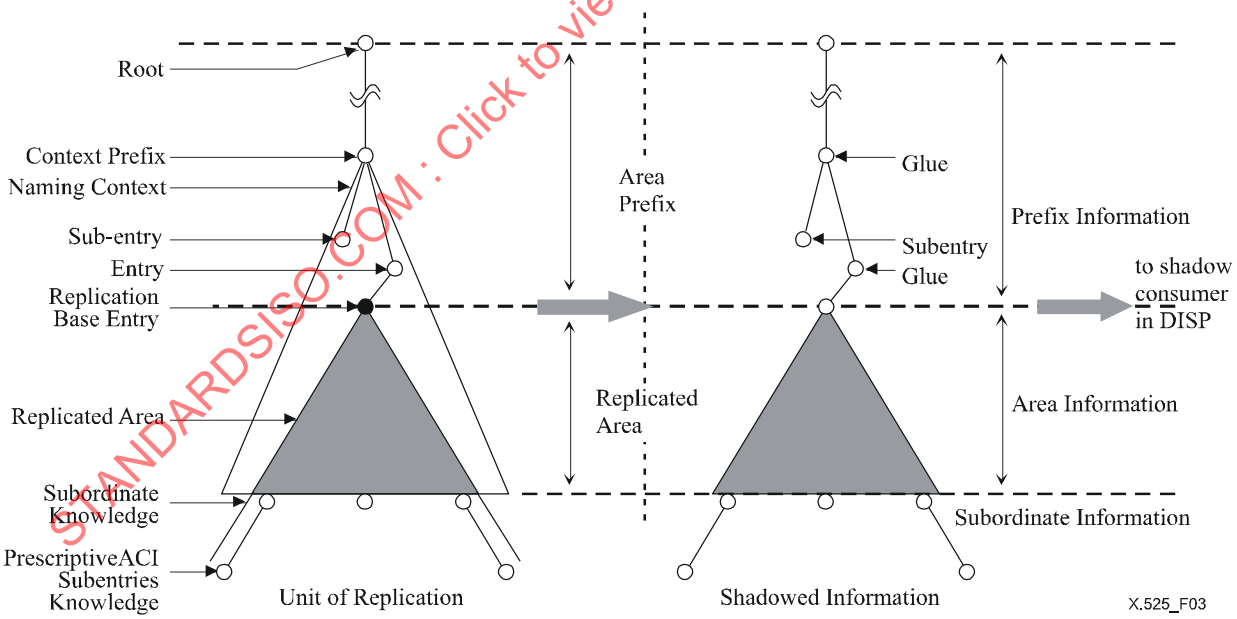

Figure 3 – Shadow supplier derivation of shadowed information

As illustrated at the left of Figure 3, the replicated area is always fully contained within a single naming context. The root of the subtree representing the replicated area is called the replication base entry. Subordinate knowledge may also be replicated. Implicit in the subordinate knowledge is the access control information which governs access to the relative distinguished name (RDN) of the subordinate knowledge. When the subordinate entry is an administrative point in another DSA, then part of this access control information may be held in **`prescriptiveACI`** in subentries beneath the subordinate knowledge. This knowledge, the refined replicated area, and the area prefix constitute the unit of replication.

This means that the specification of a unit of replication may extend beyond the naming context; however, the replicated area itself is limited to the naming context. From this unit of replication specification, the shadow supplier can derive a representation of the shadowed information, which, as shown at the right of the figure, includes the prefix information, the area information (representing information held by DSEs in the replicated area), and (optionally) subordinate information. This shadowed information is subsequently conveyed by protocol to the shadow consumer which then integrates the information into its own DSA information tree. The shadowed information is built out of shadowed DSEs (SDSEs), which are discussed in 7.2.1. The establishment of shadowed information is discussed in 7.2.2.

Figure 4 illustrates the derivation of shadowed information where extended knowledge is included.

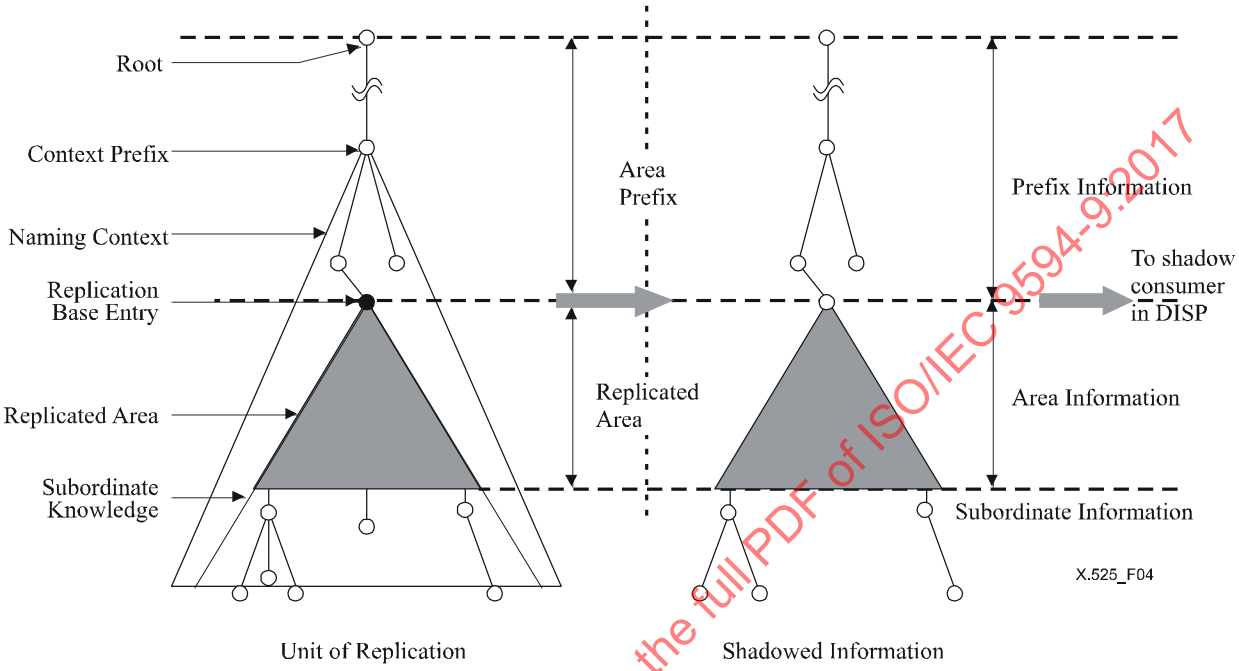

Figure 4 – Shadow supplier derivation of shadowed information with extended knowledge

7.2.1 Shadowed DSA-Specific Entries (SDSEs)

Shadowed DSA-specific entry (SDSE): That information being shadowed that is associated with a specific name. The SDSE represents the information shadowed from a DSE in the shadow supplier to a DSE in the shadow consumer, and is therefore not part of the DSA Information Model.

An SDSE is analogous to a DSE and consists of:

- SDSE type (always);
- user attributes (derived from entry information for DSEs corresponding to entries that are to be shadowed);
- operational attributes (present as required);
- subordinate-completeness flag (for area and subordinate information only);
- attribute-completeness flag (present for area information only);
- attribute-values-incomplete flag (present for area information only).

7.2.1.1 SDSE type

DSE types are defined in Rec. ITU-T X.501 | ISO/IEC 9594-2. SDSE type, as specified in 11.3.1.1, is analogous to DSE type, but has fewer relevant options: **`root`**, **`glue`**, **`cp`**, **`entry`**, **`alias`**, **`subr`**, **`nssr`**, **`admPoint`**, **`subEntry`** and **`sa`**.

7.2.1.2 Subordinate-completeness flag

The subordinate-completeness flag is a Boolean that is present for SDSEs within the area information and subordinate information. If the shadow supplier does not intend to provide information about subordinate completeness, the value **`FALSE`** is used for each SDSE. Otherwise the flag has the following semantics:

The flag is **`TRUE`** only if one of the following conditions is met for a particular SDSE:

a) it represents a leaf entry;

b) the replicated area contains SDSEs for each subordinate entry and each subordinate reference known to the master DSA, and if the SDSE represents an NSSR, this knowledge is represented in the SDSE.

The flag is **FALSE** if one of the following conditions is met for a particular SDSE:

a) the subordinates known to the master for that particular SDSE are not all present in the shadowed information;

b) in the case of a shadow supplier DSA performing secondary shadowing, if its shadow supplier had set the flag to **FALSE** or if its shadow supplier had set the flag to **TRUE** and the secondary shadow supplier chooses to set its to **FALSE**.

7.2.1.3 Attribute-completeness flag

The attribute-completeness flag is a Boolean and is **TRUE** if, and only if, all user attributes of the entry, all relevant collective attributes, all values of such user or collective attributes, and all context information associated with those values, are present for the SDSE. It is only present for SDSEs containing entry information.

The attribute-completeness flag is not used with respect to Directory operational attributes; it is always assumed that they are not all present in the SDSE.

7.2.1.4 Attribute-values-incomplete flag

The attribute-values-incomplete flag is a list of the attribute types present in the SDSE for which not all attribute values are present in the SDSE. It is only present for SDSEs containing entry information.

NOTE – Attribute values may be missing because of selective shadowing based on contexts.

7.2.2 Establishment of shadowed information

The shadowed information represents three basic types of information: prefix information, area information, and subordinate information. Each of these is discussed in the following subclauses.

7.2.2.1 Prefix information

If the replicated area does not start immediately below the root of the DIT, the shadowed information will include SDSEs for each entry that is part of the area prefix of the replicated area (the path down from the root of the DIT to, but not including, the replication base entry, and any relevant subentries). SDSEs for prefix information are constructed as shown below.

a) If the DSE is an administrative point that has attributes pertaining to the replicated area, or that has one or more associated subentries whose subtree scope includes some or all of the replicated area, the SDSE is of type **admPoint**. If the DSE is also of type **cp**, the corresponding SDSE is of additional type **cp**. Any attributes that are relevant for the replicated area are included in the SDSE. The **administrativeRole** attribute shall be included in all administrative point SDSEs which are relevant to the shadowed information.

b) For subentries below the administrative point for which the subtree scope includes some or all of the replicated area, SDSEs of type **subentry** may be included in the shadowed information. If the subtree scope of such a subentry does not include the replicated area or parts of it, no SDSE for this subentry need be included. Collective attributes, schema and access control information selected for the area information are represented in SDSEs of type **subentry**.

c) There is an empty SDSE of type **root** for the root DSE.

d) If the DSE is only of type **cp**, the SDSE is of type **cp**.

e) All other DSEs not described in a), b), c), or d) are represented as SDSEs of type **glue** and will only represent the RDN of the entry.

There are no subordinate-completeness flags in area prefix SDSEs.

7.2.2.2 Area information

All entries in the shadow supplier information tree that are included in the replicated area are represented in the shadowed information as SDSEs of type **entry** (unless removed by filtering). These SDSEs contain the attributes of the entries as selected by the attribute selection of the shadowing agreement. Collective attributes held in subentries are selected in the same manner as other attributes and are represented in SDSEs of type **subentry**. If any attributes of an entry have been selected for inclusion in the shadow, the **objectClass** attribute and the relevant entry access control information will be included in the SDSE for that entry. The attribute-completeness flag is set to indicate whether all user attributes in the DSE and all relevant collective attributes are present for the SDSE. The **collectiveExclusions** operational attribute, if present, is always included in the SDSE.

If the DSE is of type **admPoint,** the corresponding SDSE is of additional type **admPoint** and SDSEs of type **subentry** for all relevant subentries immediately subordinate to the administrative point DSE are included in the shadowed information. The rules for inclusion of subentries are stated in 7.2.2.1.

If the DSE is of type **cp**, the corresponding SDSE is of additional type **cp**.

If subordinate knowledge is specified, and if the DSE is of type **nssr**, the corresponding SDSE is of additional type **nssr** and the **nonSpecificKnowledge** attribute shall be included.

If filtering has been applied to the replicated area, the resulting shadowed information may no longer be contiguous. There may be entries that have been removed by filtering that cause the tree structure of the shadowed information to break down. For each entry that has been removed by filtering, the following rules are applied:

a) If there are SDSEs subordinate to that entry within the shadowed information that are not filtered out, an SDSE of type **glue** for the removed entry is added to the shadowed information. The subordinate completeness flag is set as specified in 7.2.1.2. As this SDSE contains no entry information, it has no attribute completeness flag. If the **entryACI** operational attribute is present and holds relevant ACI, e.g., naming, then the attribute (containing at least the relevant ACI) shall always be included in the SDSE.

b) If there are no other SDSEs subordinate to the entry within the shadowed information, the subordinate-completeness flag of the SDSE for the entry immediately superior to the removed entry is set to **FALSE** and the SDSE for the removed entry is excluded from the shadowed information.

c) If the DSE is of type **admPoint**, it is always shadowed and the **administrativeRole** attribute is included.

Each SDSE in area information has a subordinate-completeness flag. The conditions for setting this flag are specified in 7.2.1.2.

7.2.2.3 Subordinate information

The type of subordinate information required (i.e., master access points, shadow access points, or both; and whether extended knowledge is to be included or not) is specified in the shadowing agreement.

If subordinate knowledge is supplied, subordinate references directly below the replicated area (master, shadow, or both types of knowledge as appropriate) are included as SDSEs of type **subr**, complete with the appropriate knowledge and access control information.

If subordinate knowledge is supplied, and the supplying DSE (of type **subr)** is also of type **admPoint**, then the SDSE shall additionally be of type **admPoint** and the **administrativeRole** attribute shall be supplied. If such a DSE has any immediately subordinate subentries containing **prescriptiveACI** relating to the administrative point, then they shall also be supplied as SDSEs in the shadowed information.

NOTE – A DSE can be of type subr and admPoint in a superior DSA, when the naming context in the subordinate DSA is the start of a new administrative area.

If extended knowledge is specified, subordinate references below (but not immediately subordinate to) the replicated area (master, shadow, or both) are included as SDSEs of type **subr** or **nssr**, complete with the appropriate knowledge and access control information. Subordinate **glue** SDSEs shall be inserted to maintain connection with SDSEs in the replicated area. This may create **glue** SDSEs which are either within or below the replicated area. No other **glue** SDSEs are provided to support subordinate information.

If **subordinates** is specified, then the supplier shall send subordinate entries and a subordinate reference, and the SDSEs will be of type **subr**, **entry**, and **cp**. The subordinate entries shall contain attributes according to the attribute selection. In addition, if the supplying DSE is of type **admPoint**, then the SDSE shall additionally be of type **admPoint** and the **administrativeRole** attribute shall be supplied. All appropriate subentries, with only the appropriate information, below the **admPoint** DSE shall also be supplied as SDSEs in the shadowed information.

subr and **nssr** SDSEs carry a subordinate-completeness flag. **glue** SDSEs added for the purpose of extended knowledge carry no subordinate-completeness flag and are always assumed to be incomplete (with respect to subordinate knowledge).

More detailed information on the unit of replication and the representation of shadowed information is contained in 9.2.

7.3 Shadow operations

Shadowed information is transmitted from the shadow supplier to the shadow consumer by using Directory shadow operations. These operations provide two fundamentally different models for updating shadowed information:

– shadow supplier-initiated shadowing (a "push" model); and

– shadow consumer-initiated shadowing (a "pull" model).

These models are described more fully in clause 10.

In either model, the information transmitted by protocol takes one of two forms:

– *total* in which the complete set of information within the replicated area is transmitted. Each element is an SDSE;

– *incremental* in which only changes to the replicated area are transmitted. Each element is an SDSE change. SDSE changes reflect the net effect of changes that have been made to the corresponding DSEs in the replicated area since the previous update, whether these changes originally occurred as a result of changes to individual DSEs (adds, deletes, etc.) or as a result of changes to multiple DSEs (e.g., resulting from a **ModifyDN** operation).

Three shadow operations are defined. The **coordinateShadowUpdate** operation is used in the push model to enable the shadow supplier to indicate the shadowing agreement for which it intends to send an update, to indicate the time the last update was sent for that agreement, and the intended update strategy (e.g., **total** or **incremental**). If a positive result is received in response to a **coordinateShadowUpdate** operation, the shadow supplier uses the **updateShadow** operation to convey the shadowed information or the changes in the shadowed information, as indicated by the update strategy. For the pull model, the shadow consumer uses a **requestShadowUpdate** operation to indicate the shadowing agreement for which it wishes to receive an update, the time supplied in the last update for that agreement, and the desired update strategy. If the parameters of the **requestShadowUpdate** operation are acceptable to the shadow supplier, a positive result is sent to the shadow consumer. The shadow supplier uses the **updateShadow** operation to convey the shadowed information or the changes to the shadowed information, as indicated by the update strategy. These operations are described in detail in clause 11.

7.4 DSA Shadow Bind and DSA Shadow Unbind operation

The **DSAShadowBind** and **DSAShadowUnbind** operation, defined in 7.4.1 and 7.4.2 respectively, are used by a DSA at the beginning and the end of a particular period of providing shadow updates.

7.4.1 DSA Shadow Bind

A **dSAShadowBind** operation is used at the beginning of a period of providing shadows.

```
dSAShadowBind OPERATION ::= dSABind
```

The components of the **dSAShadowBind** are the same as the components of **dSABind** (see Rec. ITU-T X.518 | ISO/IEC 9594-4) with the following differences:

7.4.2 DSA Shadow Unbind

The unbinding at the end of a period of providing shadows is for the Open System Interconnection (OSI) environment specified in 7.6.4 and 7.6.5 of Rec. ITU-T X.519 | ISO/IEC 9594-5 and for the Transmission Control Protocol/Internet Protocol (TCP/IP) environment in 9.2.2 of Rec. ITU-T X.519 | ISO/IEC 9594-5.

8 Shadow operational binding

This clause defines the operational binding type for shadowing. It uses the elements and mechanisms of the DSA Operational Framework defined in Rec. ITU-T X.501 | ISO/IEC 9594-2.

The shadow operational binding type may be used to administer a shadowing agreement reached between the administrative authorities of two DSAs. Otherwise, the administration of such an agreement is outside the scope of this Directory Specification. An instance of this operational binding type creates the environment in which shadow operations can be carried out between the two DSAs. Each instance is identified by an **OperationalBindingID** also referred to as **AgreementID**. The **AgreementID** is modified in a **modifyOperationalBinding** operation.

8.1 Shadow operational binding type characteristics

8.1.1 Symmetry and roles

The shadow operational binding type is an asymmetrical type of operational binding. The two roles in a binding of this type are:

– the role of the shadow supplier (associated with the abstract role "A");

– the role of the shadow consumer (associated with the abstract role "B").

A detailed description of the concept of roles is given in Rec. ITU-T X.501 | ISO/IEC 9594-2.

8.1.2 Agreement

The agreement that has to be exchanged during the establishment of the shadow operational binding or subsequent modifications is defined by the ASN.1 type **ShadowingAgreementInfo** defined in 9.1.

8.1.3 Initiator

The establishment, modification, and termination of the shadow operational binding can be initiated by either the DSA with role shadow supplier (ROLE-A) or by the DSA with role shadow consumer (ROLE-B).

8.1.4 Establishment parameters

No additional parameters are transferred during the establishment of the operational binding.

8.1.5 Type identification

The shadow operational binding information object is identified by the value of the ID field of the class assigned as part of its definition.

8.2 DSA procedures for operational binding management

A set of operations has been defined for managing operational bindings (see Rec. ITU-T X.501 | ISO/IEC 9594-2). The use of these operations for management of a shadow operational binding is described in 8.2.1 to 8.2.3 below. These procedures apply to DSAs which support the **directoryOperationalBindingManagementAC**, as defined in Rec. ITU-T X.519 | ISO/IEC 9594-5. In the event of a protocol loss while initiating, modifying, or terminating a shadow operational binding, neither success nor failure can be assumed. It is the responsibility of the initiator to ensure both parties reach a common understanding of the state of the operation. Should the responder receive a proposal to activate a shadowing agreement with an existing ID, it shall return a **duplicateID** error as defined in Rec. ITU-T X.501 | ISO/IEC 9594-2. Procedures for management of the shadow operational binding for DSAs which do not support the **directoryOperationalBindingManagementAC** are outside the scope of this Directory Specification.

8.2.1 Establishment procedure

Once an agreement for shadowing has been made between two Administrative Authorities (using procedures outside the scope of this Directory Specification), a shadowing agreement between two DSAs is activated with an **establishOperationalBinding** operation, as defined in Rec. ITU-T X.501 | ISO/IEC 9594-2. As arguments to this operation, the initiating DSA supplies the **AgreementID** for the instance of the binding, the role of the initiating DSA for this binding instance (shadow supplier or shadow consumer), and the **ShadowingAgreementInfo**.

```
AgreementID ::= OperationalBindingID
```

AgreementID identifies the shadowing agreement being activated. It shall be unique between the pair of DSAs, and is used in subsequent operations to identify this agreement.

If other parameters are included, they are ignored.

The values for the parameters in **ShadowingAgreementInfo** are simply accepted or rejected; there is no negotiation. The responding DSA does not have the option of returning a modified set of acceptable parameter values. Assuming a successful outcome of the request to establish a shadow operational binding, the shadow supplier and shadow consumer have the same information in their shadowing agreement.

If the **establishOperationalBinding** is successful, the shadowing agreement becomes active.

Errors returned in response to an **establishOperationalBinding** operation are interpreted according to the error description in Rec. ITU-T X.501 | ISO/IEC 9594-2.

8.2.2 Modification procedure

8.2.2.1 Modification of the agreement

Modification of the parameters of a shadowing agreement is agreed as part of the agreement for shadowing. Modification of these parameters results in a new shadowing agreement being established. The parameters of the agreement may be exchanged using a **modifyOperationalBinding** operation. The DSA Administrative Authorities should consider the effect of agreement modification on any secondary shadows prior to the modification operation as these secondary agreements may be required to be modified, updated or terminated.

The modification procedure does not allow modification of the name of the replicated base entry or the DSA's roles.

The arguments to the **modifyOperationalBinding** are the **AgreementID** for this instance of the binding, the **AgreementID** for the binding after the operation has been applied, the role of the DSA for this binding instance (shadow supplier or shadow consumer), and the new **ShadowingAgreementInfo**. The values for the parameters of the **ShadowingAgreementInfo** for the modify operation are accepted or rejected; there is no negotiation. Assuming a successful outcome to the request for a modification of the shadow operational binding, the shadow consumer and shadow supplier have the same information in their shadowing agreement.

After the modification operation, the data associated with the prior agreement remains in the shadow consumer and becomes the shadowed information for the new agreement. This does not preclude the shadow consumer requesting a total refresh. An update of the shadowed information may be required to remove inconsistencies between prior shadowed data and data required to be shadowed as specified in the **UnitOfReplication** associated with the new shadowing agreement.

Errors returned in response to a **modifyOperationalBinding** operation are interpreted according to the error description in Rec. ITU-T X.501 | ISO/IEC 9594-2.

8.2.2.2 Update of secondary shadow information

Either the shadow supplier or shadow consumer may signal, with the establishment of the operational binding, that secondary shadow information should be supplied by the shadow consumer to the shadow supplier of the replicated area. The secondary shadow information indicates the set of DSAs holding commonly usable copies of the replicated areas. A DSA acting as both a shadow consumer and shadow supplier for different shadowing agreements for the same replicated area conveys in this information to its shadow supplier with a value of the **ModificationParameter**.

```
ModificationParameter ::= SEQUENCE {
  secondaryShadows  SET OF SupplierAndConsumers,
  ... }
```

secondaryShadows contains a complete set of secondary shadow DSA access points holding commonly useful copies of the replicated area.

8.2.3 Termination procedure

Termination of the operational binding deactivates the shadowing agreement. The termination is accomplished by either the shadow supplier or the shadow consumer initiating the **terminateOperationalBinding** operation as specified in Rec. ITU-T X.501 | ISO/IEC 9594-2. No additional parameters are defined for the **terminateOperationalBinding** operation. Conditions may have been specified as part of the bilateral agreement regarding subsequent treatment of the data upon termination, such as the removal of the shadowed information from the shadow consumer DSA within a specified time. Such conditions take effect upon termination. In the event that a shadow operational binding is terminated, the shadow consumer shall deactivate any secondary shadowing agreements dependent on information in the shadowing agreement in question. The deactivation of secondary shadowing agreements is independent of and typically happens sometime after the original **terminateOperationalBinding** operation.

If the **terminateOperationalBinding** is successful, the shadowing agreement ceases to be active.

Errors returned in response to a **terminateOperationalBinding** operation are interpreted according to the error description in Rec. ITU-T X.501 | ISO/IEC 9594-2.

8.2.4 Operations and procedures

The operations that can be executed in the active state of a shadow operational binding are those defined within the **shadowConsumerInitiatedAC** and **shadowSupplierInitiatedAC** application contexts defined in Rec. ITU-T X.519 | ISO/IEC 9594-5:

- **updateShadow** operation;
- **requestShadowUpdate** operation;
- **coordinateShadowUpdate** operation.

These operations are defined in clause 11. The associated service is defined in clause 10.

8.3 Operational binding

This subclause defines the shadow operational binding information object class as an instance of the class **OPERATIONAL-BINDING** as defined in Rec. ITU-T X.501 | ISO/IEC 9594-2.

```
shadowOperationalBinding OPERATIONAL-BINDING ::= {
  AGREEMENT              ShadowingAgreementInfo
  APPLICATION CONTEXTS
    {{shadowSupplierInitiatedAC
      APPLIES TO  {All-operations-supplier-initiated}} |
    {shadowConsumerInitiatedAC
      APPLIES TO  {All-operations-consumer-initiated}}}
  ASYMMETRIC
    ROLE-A { -- shadow supplier role
      ESTABLISHMENT-INITIATOR  TRUE
      ESTABLISHMENT-PARAMETER  NULL
      MODIFICATION-INITIATOR   TRUE
      TERMINATION-INITIATOR    TRUE }
    ROLE-B { -- shadow consumer role
      ESTABLISHMENT-INITIATOR  TRUE
      ESTABLISHMENT-PARAMETER  NULL
      MODIFICATION-INITIATOR   TRUE
      MODIFICATION-PARAMETER   ModificationParameter
      TERMINATION-INITIATOR    TRUE}
  ID                     id-op-binding-shadow }

All-operations-consumer-initiated OPERATION ::=
  {requestShadowUpdate | updateShadow}

All-operations-supplier-initiated OPERATION ::=
  {coordinateShadowUpdate | updateShadow}
```

The type **ShadowingAgreementInfo** is defined in 9.1.

9 Shadowing agreement

Before shadowing takes place between two DSAs, an agreement covering the terms of the shadowing is required. There may be a requirement to establish the policy covering the shadowing which can occur. Administrative Authorities may be required to configure the environment to enable the shadowing to occur, including identification of the information to be shadowed and the type of update, etc. The types of agreements required will vary depending on the environment within which the shadowing will occur. In some cases, an explicit shadowing agreement, contractual in nature, may be required. In other cases, the shadowing agreement may be implicit, based on the agreement for shadowing between Administrative Authorities of the relevant DMDs.

In addition to the parameters of a shadowing agreement (see below), this agreement for shadowing may include policy conditions for the treatment of the data upon termination of the agreement, such as for the removal of shadowed information upon termination (or modification) of the shadowing agreement itself. Administrative Authorities also need to consider factors affecting interoperability when establishing agreements.

A shadowing agreement is required before shadowed information may be shared between any pair of DSAs. This establishes the technical parameters of the agreement, specifying update frequency, replicated area and information to be shadowed.

The shadowing agreement may be activated by its inclusion in an **establishOperationalBinding** operation (as outlined in 8.2.1) or by means outside the scope of this Directory Specification. In addition, a shadowing agreement may be modified through a **modifyOperationalBinding** operation (as outlined in 8.2.2). No negotiation of parameters of the agreement is supported by the operational binding management protocol. The parameters are either accepted or rejected. A shadowing agreement may be terminated through a **terminateOperationalBinding** operation.

9.1 Shadowing agreement specification

The shadowing agreement is specified as:

```
ShadowingAgreementInfo ::= SEQUENCE {
  shadowSubject          UnitOfReplication,
  updateMode             UpdateMode DEFAULT supplierInitiated:onChange:TRUE,
  master                 AccessPoint OPTIONAL,
  secondaryShadows  [2]  BOOLEAN DEFAULT FALSE }
```

shadowSubject specifies the subtree, entries and attributes to shadow. The components of the **UnitOfReplication** are defined in 9.2.

updateMode specifies when updates of a shadowed area are scheduled to occur. The components of **UpdateMode** are defined in 9.3.

master contains the access point of the DSA containing the mastered area. This element is optional and need only be supplied for optimization purposes.

secondaryShadows permits secondary shadow information to subsequently be supplied to the shadow supplier.

9.2 Unit of replication

This subclause describes how portions of the DIT can be replicated by defining the granularity of the DIT information that can be shadowed. The unit of replication is defined within the Directory information model, and a specification mechanism is provided. The shadowing mechanism in the Directory is based on the definition of the subset of the DIT that will be shadowed. This subset is called a *unit of replication*.

Because shadowing in the Directory is only defined between pairs of DSAs, there is a constraint that the shadowed information shall be completely within a single DSA. The specification of the unit of replication may extend beyond a naming context, but the replicated area is limited to the naming context.

The unit of replication comprises a three-part specification which defines the scope of the portion of the DIT to be replicated, the attributes to be replicated within that scope, and the requirements for subordinate knowledge. The unit of replication also implicitly causes the shadowed information to include policy information in the form of operational attributes held in entries and subentries (e.g., access control information) which is to be used to correctly perform Directory operations. The policy information to be included begins at an autonomous administrative point and extends to the replication base entry, but does not include it.

The unit of replication is specified as:

```
UnitOfReplication ::= SEQUENCE {
  area                 AreaSpecification,
  attributes           AttributeSelection,
  knowledge            Knowledge OPTIONAL,
  subordinates         BOOLEAN DEFAULT FALSE,
  contextSelection     ContextSelection OPTIONAL,
  supplyContexts  [0]  CHOICE {
    allContexts         NULL,
    selectedContexts    SET SIZE (1..MAX) OF CONTEXT.&id,
    ... } OPTIONAL }

AreaSpecification ::= SEQUENCE {
  contextPrefix    DistinguishedName,
  replicationArea  SubtreeSpecification,
  ... }

Knowledge ::= SEQUENCE {
  knowledgeType       ENUMERATED {
    master (0),
    shadow (1),
    both   (2)},
  extendedKnowledge  BOOLEAN DEFAULT FALSE,
  ... }
```

area defines the replicated area. It includes the context prefix of the naming context containing the replicated area and the subtree specification relative to that context prefix. For the case where a DSA is shadowing first level knowledge from a first level DSA, the **contextPrefix** component is empty. **SubtreeSpecification** is defined in Rec. ITU-T X.501 | ISO/IEC 9594-2. The names used in **area** shall be the primary distinguished names, without context information or alternative distinguished values.

attributes defines the set of attributes to be shadowed. It includes specification of user attributes (including collective attributes) and operational attributes, as described in 9.2.2.

knowledgeType defines the knowledge references to be shadowed. It includes specification of the type of references (master/shadow) to be shadowed as well as whether the knowledge requested is extended knowledge.

master indicates that only references to master naming contexts are to be supplied.

shadow indicates that only references to commonly usable replicated areas are to be supplied.

both indicates that references to both master and shadowed naming contexts are to be supplied.

If **extendedKnowledge** is specified, then all subordinate and non-specific subordinate references of the naming context, which are subordinate to the area prefix are included in the unit of replication. To achieve this, **glue** SDSEs are included, as necessary, in the shadowed information to represent all entries between the lower boundary of the replicated area and the subordinate knowledge references.

subordinates is used to indicate that subordinate entries, rather than simply subordinate references, are to be copied to the consumer DSA. **subordinates** may only be **TRUE** if **knowledge** is requested and **extendedKnowledge** is **FALSE**.

contextSelection is used to refine further the information selection. It may be used to select which attribute values of the attributes selected in **attributes** are to be shadowed. Only attribute values selected by **contextSelection** shall be shadowed. Selection is based on the same rules as described for entry information selection in 7.6.2 of Rec. ITU-T X.511 | ISO/IEC 9594-3. **contextSelection** is also applied to the alternative distinguished values of naming attributes and so may affect the names of SDSEs (it is not applied to the primary distinguished values which are always shadowed). If **contextSelection** is not specified, then all attribute values for all attributes in **attributes** shall be shadowed (i.e., default contexts are not applied to **UnitOfReplication** as they are for **EntryInformationSelection**).

supplyContexts indicates that the shadow consumer wishes to receive context information associated with the attribute values that are selected to be replicated. If **allContexts** is specified, then all context information is supplied with the attribute values that are shadowed. If **selectedContexts** is used, then only context information of the type(s) specified is supplied with the attribute values that are shadowed. If **supplyContexts** is omitted, then the supplying DSA shall supply attribute values devoid of all context information.

supplyContexts is not applied to the distinguished attribute values shadowed as part of the SDSE name. If any alternative values are included in an **AttributeTypeAndDistinguishedValue** in an RDN in the SDSE, then the associated context list shall also be included for the primary distinguished value and all the alternative distinguished values in the **AttributeTypeAndDistinguishedValue**.

The following subclauses define the components of the unit of replication in detail. Support, by a shadow supplier DSA, for various components is optional as specified in 13.3.1 of Rec. ITU-T X.519 | ISO/IEC 9594-5.

9.2.1 Area specification

The replicated area is specified by defining a subtree of the DIT and refining that subtree to exclude those portions not required. The refinements include a filtering of entries, based on their object class. These stages are described in 9.2.1.1 and 9.2.1.2.

9.2.1.1 Subtree boundary specification

The first stage is to specify the shape of the subtree that is to be shadowed within a DSA. This is done by drawing the boundary of the subtree based on the tree structure using the subtree specification mechanism as defined in Rec. ITU-T X.501 | ISO/IEC 9594-2. The component **base** of **SubtreeSpecification** is used to provide the replication base entry of the unit of replication relative to the context prefix from which the replicated area was derived. The **chop** component of **SubtreeSpecification** is used to define the lower boundary of the subtree that is to be shadowed. The entries that can be referenced by either the **specificExclusions** or the **maximum** component are limited by the lower boundary of the naming context holding the replication base entry. If the **chop** component is absent, the unit of replication includes the whole subtree starting with the **base** and proceeding down to the lower boundary of the naming context.

The component **minimum** shall not be used to specify a subtree to be shadowed.

9.2.1.2 Subtree refinement

The next stage of refinement is to apply a filter to the selected subtree. The **specificationFilter** component of **SubtreeSpecification** is used to specify the filter. Filtering is done on object class only.

Filtering may result in a unit of replication that is no longer a connected subtree in the DSA, from the viewpoint of the Directory Information Model. For such subtrees **glue** DSEs are required to be supplied for as many entries as are needed to build a connected subtree in the shadow consumer.

9.2.2 Attribute selection

This further stage of refinement of the unit of replication specifies the attributes (user, collective and Directory operational) to be shadowed.

In addition to those specified here, access control operational attributes, **createTimestamp** and **modifyTimestamp** are always included in a unit of replication. Also, if knowledge is specified (as defined in 9.2.3), the knowledge operational attributes will be included in the shadowed information and need not be enumerated as part of this attribute selection.

The **createTimestamp** and **modifyTimestamp** shall be provided by the shadow supplier in the shadowed information (entries and subentries). The **createTimestamp** shall be conveyed in the **SDSEContent** during a total refresh or if a new shadow DSE is added. The **modifyTimestamp** shall always be conveyed in the **SDSEContent** if present in the shadow supplier's DSE for that entry or subentry.

The following attributes shall be provided by the shadow supplier in the shadowed information (entries and subentries):

- **pwdModifyEntryAllowed**
- **pwdChangeAllowed**
- **pwdMaxAge**
- **pwdExpiryAge**
- **pwdMinLength**
- **pwdVocabulary,**
- **pwdAlphabet,**
- **pwdDictionaries,**
- **pwdExpiryWarning**
- **pwdGraces**
- **pwdFailureDuration**
- **pwdLockoutDuration**
- **pwdMaxFailures**
- **pwdMaxTimeInHistory**
- **pwdMinTimeInHistory**
- **pwdHistorySlots**
- **pwdRecentlyExpiredDuration**
- **pwdEncAlg**

The following attributes shall be provided by the shadow supplier in the shadowed information (entries):

- **pwdStartTime**
- **pwdExpiryTime**
- **pwdEndTime**
- **userPwdHistory.**
- **userPwdRecentlyExpired**
- **pwdAdminSubentryList**

The attribute selection shall be specified to reflect, if at all possible, any restrictions on shadow consumer access to the information. However, it is possible that some security policies may cause very limited exceptions to this norm where particular information is withheld from the shadowed information.

The principles of attribute selection are:

a) The selection takes place within the shadow supplier DSA, in accordance with **AttributeSelection** at the time of shadowing. There are no actions whatever that are applied by the shadow consumer DSA.

b) Attributes that are to be selected for shadowing, SDSE by SDSE, can be selected on the basis of the class of the entries and/or subentries being shadowed, or for generic use within all shadowed entries.

NOTE 1 – This permits flexibility such as either shadowing the telephoneNumber attribute for all entries which have this attribute or shadowing the telephoneNumber attribute only for entries of class organizationalUnit.

```
AttributeSelection ::= SET OF ClassAttributeSelection

ClassAttributeSelection ::= SEQUENCE {
  class            OBJECT IDENTIFIER OPTIONAL,
  classAttributes  ClassAttributes DEFAULT allAttributes:NULL }
```

```
ClassAttributes ::= CHOICE {
  allAttributes  NULL,
  include        [0]  AttributeTypes,
  exclude        [1]  AttributeTypes,
  ... }

AttributeTypes ::= SET OF AttributeType
```

Each element of **AttributeSelection** is a **ClassAttributeSelection** element, and specifies the attributes that the shadow supplier is to select for shadowing. Specification of attributes for an object superclass also applies to any subclasses of the named class. If the class is omitted, the selection applies to all classes.

The default **allAttributes** specifies that all user attributes (including collective attributes) are to be included. If there are relevant collective attributes associated with the class, the appropriate **collectiveAttributeSubentrys** are implicitly included. If any Directory operational attributes (other than access control, timestamps and knowledge) are to be included, they shall be identified in the **include** element of the specification.

Attributes are implicitly included in the case where **allAttributes** is specified. In addition, when using the **exclude** specification, any attributes contained in an entry which are not explicitly excluded are implicitly included. The specification of an attribute supertype implicitly includes any subtypes of that attribute.

Explicit **include** or **exclude** of a collective attribute for a particular class results in the corresponding inclusion or exclusion of the collective attributes in the holding subentries.

Where entries belong to more than one of the specified classes, the specifications are cumulative. In the case of conflicting specifications, **include** has priority over explicitly excluded attributes and **exclude** has priority over implicitly included attributes.

NOTE 2 – If a specific collective attribute is shadowed, it may be returned by the shadow consumer as part of EntryInformation even if it has not been specifically included for that entry by AttributeSelection. This is because the AttributeSelection value is not considered by the shadow consumer in fulfilling the abstract service.

9.2.3 Subordinate knowledge

The next stage in defining the unit of replication is the inclusion of subordinate knowledge. This knowledge may include subordinate knowledge of either master or shadowed naming contexts and may include specific and/or non-specific references. Additionally, such subordinate knowledge references may be included in the unit of replication, even if they are not immediately subordinate to entries in the replicated area, in which case they are referred to as **extendedKnowledge** references. They shall still be subordinate to the **areaPrefix**.

9.2.4 Subentries

Subentries are included in the unit of replication for access control, schema, collective attributes, contexts defaults, search-rules and password policy as described below.

9.2.4.1 Access control information

It is the responsibility of the shadow supplier to provide properly transformed access control information for each item in the unit of replication. The nature of the transformation is specified as part of the shadowing agreement and may be as simple as the identity transformation.

NOTE 1 – For example, the transformation may reflect a local policy that states it is not necessary to shadow permissions related to controlling modification of the shadowed items. Such a policy is consistent with the read-only nature of shadowed information.

The following access control information shall always be shadowed:

a) the operational attribute **accessControlScheme**, for each access control specific area in the unit of replication;

b) prescriptive access controls relevant to the reading of the replicated information and found in access-control-specific or inner points or their subentries within the replicated area up to and including the first access-control-specific or autonomous administrative point encountered proceeding from the area prefix towards the root;

c) entry access controls relevant to the reading of each entry shadowed;

d) if the entry is refined out, the replacement glue SDSE shall contain the necessary access control information, e.g., reading.

The shadow consumer shall enforce access control using the shadowed access control information.

NOTE 2 – It is desirable that changes in access control policy, as expressed by ACI, should be propagated to shadowing DSAs (and other DSAs) as soon as possible. Such changes may cause (for example) the initiation of a (normal) incremental refresh

exchange to affected DSAs, without regard for any particular periodic strategy. The refresh would include (for consistency) any other updates pending to the unit of replication. A similar consideration may apply when changes are made to a groupOfUniqueNames attribute if it pertains to access control.

9.2.4.2 Schema information

The schema information required by a shadow consumer to accommodate the shadowed information in its DSA Information Tree and to satisfy Directory query operations on that shadowed information needs to be shadowed as part of the unit of replication.

The relevant operational attributes of the **subschema** subentry are always included in the unit of replication.

9.2.4.3 Entry collection information

Collective attributes are included in or excluded from the unit of replication as user attributes. If **allAttributes** is specified, all corresponding **collectiveAttributeSubentrys** are implicitly included in the unit of replication. If user attributes explicitly included in the unit of replication are collective attributes, the corresponding attributes of the **collectiveAttributeArea** are included in the unit of replication.

9.2.4.4 Search-rule information

To the extent that search-rules are required to be enforced by the shadow consumer, the **serviceAdminSubentry** subentries shall be included in the unit of replication.

9.2.4.5 Password policy information

To have the password policy enforced by the shadow consumer, the **pwdPolicySubentry** subentries shall be included in the unit of replication.

9.2.5 Principles for the use of SDSE information

The SDSE information supplied by a total or incremental refresh shall be used to generate a set of DSEs that corresponds precisely to the set of SDSEs defined by the unit of replication, with the following exceptions:

- the **DSEType** value shall become equal to the **SDSEType** value after setting the shadow bit and resetting all bits not permitted in **SDSEType**;
- the consumer DSA may create and maintain additional operational attributes for local purposes.

It is possible for such a DSE to coincide with other DSEs (i.e., having the same name). This can occur as a result of other shadowing agreements, or because the shadowed information shares common DSEs with those held as master information or as cross references by the DSA. Where such a coincidence exists, the DSA shall maintain the DSE arising from the shadowing agreement as an independent piece of information, except that later information, derived from the same master copy, can always supersede earlier information where this is detectably the case.

As an example, the naming context Q is shadowed, as Q', to the DSA holding the superior naming context. This causes the context prefix at Q to be superimposed on the subordinate reference DSE B' which points to B, the context prefix of Q. In this case, the subordinate reference DSE shall be maintained separately from the shadowed context prefix.

9.2.6 Overlapping replicated areas

A shadow consumer may optionally be involved in two or more shadowing agreements specifying overlapping replicated areas. The procedures to be followed by DSAs that do not support overlapping replicated areas are defined in 9.2.6.1. The procedures to be followed by DSAs that support overlapping replicated areas are defined in 9.2.6.2.

9.2.6.1 Procedures for DSAs not supporting overlapping replicated areas

This subclause defines the procedures to be followed by shadow consumers that do not support overlapping replicated areas.

A shadow consumer shall not engage in two or more shadowing agreements whose **UnitOfReplication** specifies overlapping replicated areas. However, the shadow consumer may encounter cases where non-overlapping replicated areas share prefix or other information, resulting in overlapping area prefix SDSEs. A similar situation occurs when an SDSE overlaps mastered information. Thus, any **subentry** SDSEs within prefix information may be subject to separate (uncoordinated) updates from different shadowing agreements. Changes in subentries (such as prescriptive access control information) need to be associated with particular data, and updates reflecting such changes will only be sent for relevant shadowing agreements. Subentries and administrative entries for shadowing agreements which share prefix or other information with DSEs from other sources (e.g., master information or other shadowing agreements) need to be logically maintained separately and associated with the appropriate unit of replication.

9.2.6.2 Procedures for DSAs supporting overlapping replicated areas

This subclause defines the procedures to be followed by shadow consumers that support overlapping replicated areas.

Each replicated area (associated with a shadowing agreement) shall be represented in the shadow consumer by a separate "information plane". When the shadowed information associated with a shadowing agreement is updated, only the "information plane" that represents that shadowed information shall be affected.

When performing a Directory interrogation operation on a given replicated area, a shadow consumer shall do one of the following:

a) Select an "information plane" that is capable of satisfying the specified Directory operation. The procedure used to select the appropriate "information plane" is outside the scope of this Directory Specification. Once an appropriate "information plane" is found, only the **shadow** DSEs contained in that "plane" are considered during the execution of the Directory operation, i.e., information contained in other "information planes" is ignored.

b) Consider the aggregate of shadowed information the shadow consumer holds for the relevant replicated area by merging the **shadow** DSEs from different "information planes" into one single set of **shadow** DSEs, one for each replicated entry. If the resulting shadowed information is capable of satisfying the Directory operation, execute the latter on the resulting set of **shadow** DSEs.

NOTE – A shadow DSE resulting from the union of all shadow DSEs representing a given replicated entry should contain the most current shadowed information from the set of all applicable "information planes".

9.3 Update mode

The **updateMode** argument in the shadowing agreement specifies when updates are expected to occur for shadowed information.

```
UpdateMode ::= CHOICE {
  supplierInitiated  [0]  SupplierUpdateMode,
  consumerInitiated  [1]  ConsumerUpdateMode,
  ... }

SupplierUpdateMode ::= CHOICE {
  onChange    BOOLEAN,
  scheduled   SchedulingParameters,
  ... }

ConsumerUpdateMode ::= SchedulingParameters
```

The components of **updateMode** are defined in 9.3.1 through 9.3.3.

For each shadowing agreement, a choice has to be made between the shadow supplier or shadow consumer initiating the update. This is specified by selecting **supplierInitiated** or **consumerInitiated**. This choice does not preclude either party in a shadowing agreement from initiating (or attempting to initiate) an update at times outside those specified by the **updateMode**.

If **rule-based-access-control** is in place, the clearance of the peer DSA needs to be checked against the label for any shadowed attribute value to check that the peer DSA has the clearance to access the data.

9.3.1 Supplier update mode

In **SupplierUpdateMode, onChange** indicates that the shadow supplier is expected to provide updates when changes occur within the replicated area as specified by the unit of replication. Should the shadow consumer be unavailable, the shadow supplier shall resend the update within an appropriate, locally-defined, time period. If, due to the unavailability of the shadow consumer, a number of changes are outstanding, the shadow supplier may transmit them within one single **updateShadow** operation.

scheduled allows updates from the shadow supplier to be scheduled as specified by **SchedulingParameters**.

9.3.2 Consumer update mode

In **ConsumerUpdateMode** the scheduling of the update requests is as specified by the **SchedulingParameters**.

9.3.3 Scheduling parameters

The **SchedulingParameters** provide the information required to schedule the requests for updates.

```
SchedulingParameters ::= SEQUENCE {
```

```
  periodic     PeriodicStrategy OPTIONAL, -- shall be present if othertimes
  --                                      is set to FALSE
  othertimes   BOOLEAN DEFAULT FALSE,
  ... }
```

Scheduling can be based on a periodic basis (**periodic**), an exception basis (**othertimes**) or a combination of both.

If present, **periodic** indicates that update windows are expected to occur on a regular basis. **PeriodicStrategy** is used to specify the windows by providing a start time of the first window, the size of each window, and the amount of time between windows. These parameters provide guidance as to when updates are expected to occur; however, updates may also be attempted, for a number of reasons, outside the windows specified.

```
PeriodicStrategy ::= SEQUENCE {
  beginTime       Time OPTIONAL,
  windowSize      INTEGER,
  updateInterval  INTEGER,
  ... }

Time ::= GeneralizedTime
     -- as per 46.3 b) and c) of Rec. ITU-T X.680 | ISO/IEC 8824-1
```

beginTime specifies the start time of the first window.

windowSize is the length of the update window in seconds.

updateInterval is the interval between the start of one update window and the start of the next update window. The interval is expressed in seconds.

If **beginTime** is not specified, the update strategy starts at the time the shadowing agreement is activated.

othertimes indicates that updates can be scheduled according to local requirements. When this is set as part of the shadowing agreement, then the shadow supplier may include the **updateWindow** parameter during shadow update operations to signal the window for the next expected update.

If **periodic** is present and **othertimes** is **TRUE**, a window selected by **UpdateWindow** in an **updateShadow** operation, or as a result of a **coordinateShadowUpdate** or **requestShadowUpdate** operation, has precedence over those specified in **PeriodicStrategy** (e.g., if **othertimes** calls for a later time than the next periodic update according to **PeriodicStrategy**), the **PeriodicStrategy** time is ignored.

10 Directory information shadow service

The Directory information shadow service defined here provides the Directory with a mechanism to provide and support replicated information. The use of shadowed information to satisfy Directory requests is described in Rec. ITU-T X.518 | ISO/IEC 9594-4.

Once a shadowing agreement has been activated, shadowing may take place in the form of updates by using operations of the Directory Information Shadowing Protocol (DISP). Three distinct operations are available: **coordinateShadowUpdate**, **updateShadow** and **requestShadowUpdate**. Descriptions of how these operations are used for shadow supplier initiated update and for shadow consumer initiated update are provided in 10.1 and 10.2 below. In both cases, the updates for a particular agreement are sent in a single operation. The operations themselves are defined in clause 11 and the associated errors in clause 12.

10.1 Shadow supplier initiated service

This subclause describes shadow supplier initiated update using the **coordinateShadowUpdate** and **updateShadow** operations. The **coordinateShadowUpdate** operation, invoked by the shadow supplier, identifies the shadowing agreement for which the shadow supplier intends to send an update.

Upon receipt of a positive acknowledgement, the shadow supplier sends the update for the shadowing agreement by using the **updateShadow** operation.

Otherwise the shadow consumer responds with a **shadowError**. The circumstances under which particular errors will be returned are defined in clause 11.

Although the **coordinateShadowUpdate** operation applies to only a single shadowing agreement, several shadowing agreements can be updated within a single application-association. For any one shadowing agreement, the **coordinateShadowUpdate** operation (request and result) shall precede the **updateShadow** operation. Only one instance of the **updateShadow** operation can be invoked per **coordinateShadowUpdate** instance. For any one

shadowing agreement, there can only be a single **coordinateShadowUpdate** operation for which the response and **updateShadow** operation are outstanding at any one time.

Under certain circumstances, a failure of underlying services may be detected by the shadow supplier and/or shadow consumer (e.g., as a result of an OSI/IDM reject or an abort). If such an indication is received at any point prior to receipt of a positive response to the **updateShadow** operation, the shadow supplier shall assume that the combination of **coordinateShadowUpdate** and **updateShadow** failed. If the shadow consumer receives such an indication at any point prior to responding to the **updateShadow** operation, the shadow consumer shall also assume that the entire combination failed. Assuming such a failure, the shadow consumer, upon receipt of another **coordinateShadowUpdate** operation for this shadowing agreement, shall disregard any previously outstanding **coordinateShadowUpdate** rather than return an error. Procedures for recovery are outside the scope of this Directory Specification.

10.2 Shadow consumer initiated service

This subclause describes shadow consumer initiated update using the **requestShadowUpdate** and **updateShadow** operations. The **requestShadowUpdate** operation, invoked by the shadow consumer, identifies the shadowing agreement for which the shadow consumer wishes to receive an update.

If the parameters in the **RequestShadowUpdateArgument** are acceptable to the shadow supplier, a result will be returned although no information will be conveyed with it. The shadow supplier sends the update for the shadowing agreement using the **updateShadow** operation.

Otherwise the shadow supplier responds with a **shadowError**. The circumstances under which particular errors will be returned are defined in clause 11.

Although the **requestShadowUpdate** operation applies to only a single shadowing agreement, several shadowing agreements can be updated within a single application-association. For any one shadowing agreement, the **requestShadowUpdate** operation (request and result) shall precede the **updateShadow** operation. Only one instance of the **updateShadow** operation can be invoked per **requestShadowUpdate** instance. For any one shadowing agreement, there can only be a single **requestShadowUpdate** operation for which the response and **updateShadow** operation are outstanding at any one time.

Under certain circumstances, a failure of underlying services may be detected by the shadow supplier and/or shadow consumer (e.g., as a result of an OSI/IDM reject or an abort). If such an indication is received at any point prior to receipt of a positive response to the **updateShadow** operation, the shadow supplier shall assume that the combination of **requestShadowUpdate** and **updateShadow** failed. If the shadow consumer receives such an indication at any point prior to responding to the **updateShadow** operation, the shadow consumer shall also assume that the entire combination failed. Assuming such a failure, the shadow supplier, upon receipt of another **requestShadowUpdate** operation for this shadowing agreement shall disregard any previously outstanding **requestShadowUpdate** rather than return an error. Procedures for recovery are outside the scope of this Directory Specification.

11 Shadow operations

The operations of the Directory Information Shadowing Protocol (DISP), used by shadow suppliers and shadow consumers to realize the Directory information shadow service described in clause 10, are defined in 11.1 through 11.3. The associated errors are defined in clause 12.

11.1 Coordinate Shadow Update operation

The **coordinateShadowUpdate** operation is used by the shadow supplier to indicate the shadowing agreement for which it intends to send updates. The arguments of the operation may be signed (see 7.14 of Rec. ITU-T X.511 | ISO/IEC 9594-3) by the shadow supplier. If the **target** component of the **SecurityParameters** (see 7.10 of Rec. ITU-T X.511 | ISO/IEC 9594-3) in the request is set to **signed**, the shadow consumer may sign a possible result. Otherwise, the result shall not be signed.

```
coordinateShadowUpdate OPERATION ::= {
  ARGUMENT  CoordinateShadowUpdateArgument
  RESULT    CoordinateShadowUpdateResult
  ERRORS    {shadowError}
  CODE      id-opcode-coordinateShadowUpdate
}

CoordinateShadowUpdateArgument ::=
  OPTIONALLY-PROTECTED { CoordinateShadowUpdateArgumentData }
```

```
CoordinateShadowUpdateArgumentData ::= [0]  SEQUENCE {
  agreementID          AgreementID,
  lastUpdate           Time OPTIONAL,
  updateStrategy       CHOICE {
    standard             ENUMERATED {
      noChanges    (0),
      incremental  (1),
      total        (2),
      ...},
    other                EXTERNAL,
    ...},
  securityParameters   SecurityParameters OPTIONAL,
  ...}

CoordinateShadowUpdateResult ::= CHOICE {
  null          NULL,
  information   OPTIONALLY-PROTECTED{ CoordinateShadowUpdateResultData },
  ...}

CoordinateShadowUpdateResultData ::= [0]  SEQUENCE {
  agreementID   AgreementID,
  lastUpdate    Time OPTIONAL,
  ...,
  ...,
  COMPONENTS OF CommonResultsSeq }
```

11.1.1 Coordinate Shadow Update parameters

The various parameters have the following meaning.

The **agreementID** argument identifies the shadowing agreement as defined in 9.1.

The **lastUpdate** argument indicates the time provided by the shadow supplier in the most recent successful update. It shall be absent if there has been no previous successful update for the shadowing agreement, or if the shadow consumer requires a full update even if there have been no changes to the shadowed information, e.g., to recover from errors.

The **updateStrategy** argument identifies the update strategy the shadow supplier intends to use for this update. Within the choice of **standard**, the shadow supplier may select **noChanges** (indicating no modifications to the shadowed information), **incremental** (indicating incremental changes), or **total** (indicating a complete replacement of the unit of replication).

The option **noChanges** should only be used when the shadow supplier wishes to inform the shadow consumer that no modifications have occurred to the replicated area since the last update (e.g., in the case where a regularly scheduled update is expected). This shall be followed by an **updateShadow** operation with **RefreshInformation** set to **noRefresh**.

The **securityParameters** argument is defined in 7.10 of Rec. ITU-T X.511 | ISO/IEC 9594-3. The **target** parameter value is set to **none**. The **securityParameters** argument shall be included if the argument is to be signed by the shadow supplier.

11.1.2 Coordinate Shadow Update success

Should the request succeed, a result shall be returned. If the result is to be signed by the shadow consumer, the **SecurityParameters** (see 7.10 of Rec. ITU-T X.511 | ISO/IEC 9594-3) component of **CommonResultsSeq** (see 7.4 of Rec. ITU-T X.511 | ISO/IEC 9594-3) shall be included in the result. If the result is not to be signed by the shadow consumer, no information shall be conveyed with the result.

11.1.3 Coordinate Shadow Update failure

Should the request fail, a **shadowError** shall be reported. Circumstances under which the particular shadow problems will be returned are defined below.

An **invalidAgreementID** shadow problem is returned if the shadow consumer DSA does not recognize the **AgreementID** specified within the set of **AgreementIDs** with this shadow supplier DSA.

An **inactiveAgreement** shadow problem is returned if the shadow consumer DSA recognizes the **AgreementID** as a valid **AgreementID** for this shadow supplier DSA, but the shadow consumer DSA understands that the **AgreementID** is inactive.

An **unsupportedStrategy** shadow problem is returned if the shadow consumer DSA does not support the refresh strategy selected by the shadow supplier DSA for this shadowing agreement.

A **missedPrevious** shadow problem is returned if the shadow consumer DSA's understanding of the time of last update is earlier than the time indicated by the value received in **lastUpdate**.

A **fullUpdateRequired** shadow problem is returned by the shadow consumer DSA to inform the shadow supplier that a total refresh is required to bring the shadow consumer DSA into a state of consistency with the shadow supplier. This could be returned, for instance, if the shadow consumer DSA is recovering from a major failure and does not currently understand its state of consistency with respect to the shadow supplier.

An **unwillingToPerform** shadow problem is returned by the shadow consumer DSA to indicate that it is unwilling to perform the update operation associated with this coordinate operation. Interpretation of this shadow problem is outside the scope of this Directory Specification.

An **unsuitableTiming** shadow problem is returned if the shadow consumer DSA is unwilling to perform the update associated with this operation at this time.

An **updateAlreadyReceived** shadow problem is returned if the shadow consumer DSA's understanding of the time of last update is later than the time indicated by the value received in **lastUpdate**.

The **invalidInformationReceived** shadow problem is not returned in response to this operation.

An **invalidSequencing** shadow problem is returned to signal the receipt of multiple consecutive **coordinateShadowUpdate** requests for a **single** shadowing agreement without completing an intervening **updateShadow** operation or receiving an underlying service failure indication.

11.2 Request Shadow Update operation

A **requestShadowUpdate** operation is used by the shadow consumer to request updates from the shadow supplier. The arguments of the operation may be signed (see 17.3 of Rec. ITU-T X.501 | ISO/IEC 9594-2) by the shadow consumer. If the **target** component of the **SecurityParameters** (see 7.10 of Rec. ITU-T X.511 | ISO/IEC 9594-3) in the request is set to **signed** and a result is to be returned, the result may be signed. Otherwise, the result shall not be signed.

```
requestShadowUpdate OPERATION ::= {
  ARGUMENT  RequestShadowUpdateArgument
  RESULT    RequestShadowUpdateResult
  ERRORS    {shadowError}
  CODE      id-opcode-requestShadowUpdate
}

RequestShadowUpdateArgument ::= OPTIONALLY-PROTECTED { RequestShadowUpdateArgumentData }

RequestShadowUpdateArgumentData ::= [0]  SEQUENCE {
  agreementID         AgreementID,
  lastUpdate          Time OPTIONAL,
  requestedStrategy   CHOICE {
    standard  ENUMERATED {
      incremental (1),
      total       (2),
      ...},
    other     EXTERNAL,
    ...},
  securityParameters  SecurityParameters OPTIONAL,
  ...}

RequestShadowUpdateResult ::= CHOICE {
  null         NULL,
  information OPTIONALLY-PROTECTED{ RequestShadowUpdateResultData },
  ...
  }

RequestShadowUpdateResultData ::= [0]  SEQUENCE {
  agreementID  AgreementID,
  lastUpdate   Time OPTIONAL,
  ...,
  ...,
  COMPONENTS OF CommonResultsSeq }
```

11.2.1 Request Shadow Update parameters

The various parameters have the following meaning.

The **agreementID** argument identifies the shadowing agreement as defined in 9.1.

The **lastUpdate** argument is the time provided by the shadow supplier in the most recent successful update. It shall be absent if there has been no previous successful update for the shadowing agreement, or if the shadow consumer requires a full update even if there have been no changes to the shadowed information, e.g., to recover from errors.

The **requestedStrategy** argument identifies the type of update being requested by the shadow consumer.

The shadow consumer may request either an **incremental** or a **total** update from the shadow supplier. However, if the shadow consumer requests an **incremental** update and the shadow supplier determines that it needs to send a **total** update, it will return a **shadowError** with **problem** set to **fullUpdateRequired**.

The **securityParameters** argument is defined in 7.10 of Rec. ITU-T X.511 | ISO/IEC 9594-3. The **target** parameter value is set to **none**. The **securityParameters** argument shall be included if the argument is to be signed by the shadow consumer.

11.2.2 Request Shadow Update success

Should the request succeed, a result shall be returned. If the result is to be signed by the shadow supplier, the **SecurityParameters** (see 7.10 of Rec. ITU-T X.511 | ISO/IEC 9594-3) component of **CommonResultsSeq** (see 7.4 of Rec. ITU-T X.511 | ISO/IEC 9594-3) shall be included in the result. If the result is not to be signed by the shadow supplier, no information shall be conveyed with the result.

The **lastUpdate** argument indicates the shadow supplier's understanding of the time at which the last update for this agreement was sent and is the time as provided by the shadow supplier DSA. This argument may only be omitted before the first instance of a **ShadowUpdate** operation for a particular shadowing agreement.

11.2.3 Request Shadow Update failure

Should the request fail, a **shadowError** shall be reported. Circumstances under which the particular shadow problems will be returned are defined below.

An **invalidAgreementID** shadow problem is returned if the shadow supplier DSA does not recognize the **AgreementID** specified within the set of **AgreementIDs** with this shadow consumer DSA.

An **inactiveAgreement** shadow problem is returned if the shadow supplier DSA recognizes the **AgreementID** as a valid **AgreementID** for this shadow consumer DSA, but the shadow supplier DSA understands that the **AgreementID** is inactive.

An **unsupportedStrategy** shadow problem is returned if the shadow supplier DSA does not support the refresh strategy selected by the shadow consumer DSA for this shadowing agreement.

A **fullUpdateRequired** shadow problem is returned by the shadow supplier DSA to inform the shadow consumer that a total refresh is required to bring the shadow consumer DSA into a state of consistency with the shadow supplier. This could be returned, for instance, if the shadow supplier DSA is unable to construct a meaningful incremental update with respect to the value received in **lastUpdate**.

An **unwillingToPerform** shadow problem is returned by the shadow supplier DSA to indicate that it is unwilling to perform the update operation associated with this request operation. Interpretation of this shadow problem is outside the scope of this Directory Specification.

An **unsuitableTiming** shadow problem is returned if the shadow supplier DSA is unwilling to perform the update associated with this request operation at this time.

The **invalidInformationReceived**, **missedPrevious**, and **updateAlreadyReceived** shadow problems are not returned in response to this operation.

An **invalidSequencing** shadow problem is returned to signal the receipt of multiple consecutive **requestShadowUpdate** requests for a single shadowing agreement without completing an intervening **updateShadow** operation or receiving an underlying service failure indication.

11.3 Update Shadow operation

An **updateShadow** operation is invoked by the shadow supplier to send updates to the shadow consumer for a unit of replication. Prior to this operation being initiated, a **coordinateShadowUpdate** or a **requestShadowUpdate** operation shall have been successfully completed for the identified shadowing agreement. The arguments of the operation may be signed (see 17.3 of Rec. ITU-T X.501 | ISO/IEC 9594-2) by the shadow supplier. If the **target** component of the **SecurityParameters** (see 7.10 of Rec. ITU-T X.511 | ISO/IEC 9594-3) in the request is set to **signed** and a result is to be returned, the result may be signed. Otherwise, the result shall not be signed.

```
updateShadow OPERATION ::= {
  ARGUMENT  UpdateShadowArgument
  RESULT    UpdateShadowResult
  ERRORS    {shadowError}
  CODE      id-opcode-updateShadow }

UpdateShadowArgument ::= OPTIONALLY-PROTECTED {UpdateShadowArgumentData }

UpdateShadowArgumentData ::= [0]  SEQUENCE {
  agreementID          AgreementID,
  updateTime           Time,
  updateWindow         UpdateWindow OPTIONAL,
  updatedInfo          RefreshInformation,
  securityParameters   SecurityParameters OPTIONAL,
  ...}

UpdateShadowResult ::= CHOICE {
  null          NULL,
  information OPTIONALLY-PROTECTED{ UpdateShadowResultData },
  ...}

UpdateShadowResultData ::= [0]  SEQUENCE {
  agreementID  AgreementID,
  lastUpdate   Time OPTIONAL,
  ...,
  ...,
  COMPONENTS OF CommonResultsSeq }
```

11.3.1 Update Shadow parameters

The various parameters have the following meaning.

The **agreementID** identifies the shadowing agreement that has been established.

The **updateTime** argument is supplied by the shadow supplier. This time is used during the next **coordinateShadowUpdate** or **requestShadowUpdate** to ensure that the shadow supplier and shadow consumer have a common view of the shadowed information.

The **updateWindow** argument, when present, indicates the next window during which the shadow supplier expects to send an update. This parameter is only allowed if the **SchedulingParameter** of the **UpdateMode** of the shadowing agreement has the **othertimes** parameter set to **TRUE.**

```
UpdateWindow ::= SEQUENCE {
  start  Time,
  stop   Time,
  ...}
```

The **updatedInfo** argument provides the information required by the shadow consumer to update its shadowed information. This may be a total copy of the shadowed information or only incremental updates for a set of SDSEs. Although this need not provide a "mirror image" in the shadow consumer of the shadow supplier's information at any particular instant in time, the updates sent shall be internally consistent for the replicated area.

The semantics of the information conveyed in this parameter shall result in the shadow consumer reflecting the changes supplied. Furthermore, each update shall be applied independently and without regard to previously transmitted updates. If for instance, a particular add or delete was sent twice (in two separate updates with different update times), the shadow consumer would not signal an error, as the effect of adding the same **shadow** DSE twice in immediate succession is the same as adding it once. Similarly, deleting twice in immediate succession is the same as deleting once. However, neither would the shadow consumer disregard the second update on the basis of having received an earlier identical update, since intervening changes to the DSE (within the update window) could make the second update significant.

The **securityParameters** argument is defined in 7.10 of Rec. ITU-T X.511 | ISO/IEC 9594-3. The **target** parameter value is set to **none**. The **SecurityParameters** argument shall be included if the argument is to be signed by the shadow supplier.

```
RefreshInformation ::= CHOICE {
  noRefresh       NULL,
  total           [0]  TotalRefresh,
  incremental     [1]  IncrementalRefresh,
  otherStrategy        EXTERNAL,
  ...}
```

noRefresh indicates that there have been no changes to the shadowed information from the previous instance to the present. This may be used where an **updateShadow** operation shall be supplied at a certain interval defined in the shadowing agreement (**updateMode**), but no modification has actually occurred. It shall not be used where the **updateShadow** operation is in response to a **coordinateShadowUpdate** or **refreshShadowUpdate** operation in which the **lastUpdate** argument has been omitted.

total provides a new instance of the shadowed information.

incremental provides, instead of a complete replacement of the shadowed information, only the changes which have occurred to that shadowed information between **lastUpdate** in the most recent **coordinateShadowUpdate** (or **requestShadowUpdate** request), and **updateTime** in the current **updateShadow** request (or **requestShadowUpdate** response).

otherStrategy provides the ability to send updates by mechanisms outside the scope of this Directory Specification.

11.3.1.1 Total refresh

The complete shadowed information is included starting at the root of the DIT and including all SDSEs within the shadowed information.

```
TotalRefresh ::= SEQUENCE {
  sDSE     SDSEContent OPTIONAL,
  subtree  SET SIZE (1..MAX) OF Subtree OPTIONAL,
  ...}

SDSEContent ::= SEQUENCE {
  sDSEType           SDSEType,
  subComplete        [0]  BOOLEAN DEFAULT FALSE,
  attComplete        [1]  BOOLEAN OPTIONAL,
  attributes         SET OF Attribute{{SupportedAttributes}},
  attValIncomplete   SET OF AttributeType DEFAULT {},
  ...}

SDSEType ::= DSEType

Subtree ::= SEQUENCE {
  rdn  RelativeDistinguishedName,
  COMPONENTS OF TotalRefresh,
  ...}
```

Absence of objects (SDSEs) formerly contained in the shadowed information indicates their deletion.

sDSEType indicates the type of DSE being shadowed. If the bits **supr, xr, shadow, immSupr** or **rhob** are set, they are ignored.

subtree is omitted for SDSEs which have no subordinate SDSEs. The RDNs used in **subtree** shall be primary RDNs, and shall include context information and all alternative distinguished values in the **valuesWithContext** component, unless the specific shadowing agreement includes a context selection which reduces the number of alternative distinguished values which are shadowed.

subComplete is a boolean that, if present, indicates whether or not subordinate knowledge is complete. If **TRUE**, subordinate knowledge is complete. If **FALSE**, subordinate knowledge is incomplete or unknown.

attComplete is a Boolean and is **TRUE** if and only if all user attributes of the entry, all values of such user attributes, and all context information associated with those values, are present for the entry. If **FALSE**, some user attributes or values or context information have been omitted. If absent, it is undefined whether or not all user attributes or values or context information are present.

attributes is comprised of all user and operational attributes specified in the shadowing agreement.

attValIncomplete is a list of those attribute types present in **attributes** for which not all attribute values are included. Attribute values may have been omitted due to context-based selection. For any attribute type listed, some attribute values have been omitted. For any attribute type not listed, all attribute values are included. **AttValIncomplete** shall not contain any attribute types that do not appear in **attributes**.

11.3.1.2 Incremental refresh

Only the changes to the shadowed information are included in the **IncrementalRefresh**.

```
IncrementalRefresh ::= SEQUENCE OF IncrementalStepRefresh

IncrementalStepRefresh ::= SEQUENCE {
  sDSEChanges
    CHOICE {add     [0]  SDSEContent,
            remove  NULL,
            modify  [1]  ContentChange,
            ...} OPTIONAL,
  subordinateUpdates  SEQUENCE SIZE (1..MAX) OF SubordinateChanges OPTIONAL }

ContentChange ::= SEQUENCE {
  rename
    CHOICE {newRDN  RelativeDistinguishedName,
            newDN   DistinguishedName} OPTIONAL,
  attributeChanges
    CHOICE {replace  [0]  SET SIZE (1..MAX) OF Attribute{{SupportedAttributes}},
            changes  [1]  SEQUENCE SIZE (1..MAX) OF EntryModification} OPTIONAL,
  sDSEType           SDSEType,
  subComplete        [2]  BOOLEAN DEFAULT FALSE,
  attComplete        [3]  BOOLEAN OPTIONAL,
  attValIncomplete   SET OF AttributeType DEFAULT {},
  ... }

SubordinateChanges ::= SEQUENCE {
  subordinate  RelativeDistinguishedName,
  changes      IncrementalStepRefresh,
  ... }
```

The sequence of **incrementalStepRefresh** within the **IncrementalRefresh** element shall be applied to the replicated area in the order supplied. This is required to support incremental updates in the case of reuse of a Distinguished Name.

incrementalStepRefresh specifies a group of changes to be applied to the replicated area.

sDSEChanges indicates changes which need to be reflected in the shadowed information.

add provides a copy of a complete SDSE. The **shadow** DSE in the shadow consumer has no subordinates. If a **shadow** DSE with this name already exists in the shadow consumer, any subordinates are deleted and the **shadow** DSE replaced.

remove indicates that this SDSE, and any subordinates to it, should not be represented by **shadow** DSEs in the shadow consumer.

modify includes those changes that need to be reflected in a particular SDSE, including the addition of new attribute values and the deletion of old attribute values.

rename is used to indicate changes to the name of the corresponding DSE. If only the RDN of an entry is changed, then the **newRDN** component is used to indicate the distinguished values of one or more attributes which need to be reflected in the SDSE. If a subtree is moved to a new parent, then the **newDN** component is used to indicate the new name which needs to be reflected in the shadow consumer's DSA information tree. **rename** is not used to add or delete attribute values. The RDNs used in **newRDN** and **newDN** shall be primary RDNs, and shall include context information and all alternative distinguished values, unless the specific shadowing agreement includes a context selection which reduces the number of alternative distinguished values which are shadowed.

If the changes to the SDSE are extensive, a complete replacement of content is achieved using **replace**. Otherwise, **changes** is used to indicate changes which need to be reflected in the SDSE.

If **attComplete** is absent, this indicates that its value is undefined and it should not be included in the SDSE.

attValIncomplete is a list of those attribute types present in the SDSE for which not all attribute values are present after the changes in this refresh have been applied. For any attribute type listed, some attribute values are not present.

For any attribute type not listed, all attribute values are present. **AttValIncomplete** shall not contain any attribute types that do not appear in the changed SDSE.

subordinateUpdates specifies a sequence of **SubordinateChanges** which shall be applied to the replicated area in the order supplied. This ordering may be used, for example, to support incremental updates in the case of reuse of a distinguished name. Each of the **SubordinateChanges** specifies changes to make to subordinates of the entry. Note that other changes to the same subordinates may be specified in other **IncrementalStepRefresh** components of **IncrementalRefresh**.

SubordinateChanges is used to indicate changes to subordinate SDSEs. The RDNs used in **subordinate** shall be primary RDNs, and shall include context information and all alternative distinguished values in the **valuesWithContext** component, unless the specific shadowing agreement includes a context selection which reduces the number of alternative distinguished values which are shadowed.

11.3.2 Update Shadow success

Should the request succeed, a result shall be returned. If the result is to be signed by the shadow consumer, the **SecurityParameters** (see 7.10 of Rec. ITU-T X.511 | ISO/IEC 9594-3) component of **CommonResultsSeq** (see 7.4 of Rec. ITU-T X.511 | ISO/IEC 9594-3) shall be included in the result. If the result is not to be signed by the shadow consumer, no information shall be conveyed with the result.

The **lastUpdate** argument is the time provided by the shadow supplier of the previous successful update. This argument may only be omitted in the first instance of a **ShadowUpdate** operation for particular shadowing agreement.

11.3.3 Update Shadow failure

Should the request fail, a **shadowError** shall be reported. Circumstances under which the particular shadow problems will be returned are defined below.

An **invalidAgreementID** shadow problem is returned if the shadow consumer DSA does not recognize the **AgreementID** specified within the list of **AgreementIDs** with this shadow supplier DSA.

An **inactiveAgreement** shadow problem is returned if the shadow consumer DSA recognizes the **AgreementID** as a valid **AgreementID** for this shadow supplier DSA, and if the shadow consumer DSA understands that the **AgreementID** is inactive.

An **invalidInformationReceived** shadow problem is returned if the shadow consumer DSA determines that, as a result of an error in the received data, it may not be able to use the received data to provide Directory services to the Directory users. As a general rule, extraneous data (e.g., entries that should have been filtered as a result of object class selection, attributes that should have been filtered out, etc.) is not considered sufficiently serious to require the return of this shadow problem as it can be ignored by the shadow consumer. Interpretation of this shadow problem is outside the scope of this Directory Specification.

An **unwillingToPerform** shadow problem is returned by the shadow consumer DSA to indicate that the shadow consumer DSA is unwilling to perform this update operation. It may be returned, for example, to indicate that the APDU size exceeds local limits. Interpretation of this shadow problem is outside the scope of this Directory Specification.

The **unsupportedStrategy**, **missedPrevious**, **fullUpdateRequired**, **unsuitableTiming**, and **updateAlreadyReceived** shadow problems are not returned in response to this operation.

An **invalidSequencing** shadow problem is returned to signal the receipt of an **updateShadow** operation for which there had been no prior **coordinateShadowUpdate** or **requestShadowUpdate** operation.

12 Shadow error

For any of the operations defined in clause 11, a **shadowError** may be returned, indicating the nature of the **ShadowProblem**, and optionally the **lastUpdate** with a more suitable **updateWindow**. If the parameters of the operation were signed (see 7.10 of Rec. ITU-T X.511 | ISO/IEC 9594-3) by the requesting DSA, then the responding DSA may sign the error parameters.

```
shadowError ERROR ::= {
  PARAMETER OPTIONALLY-PROTECTED-SEQ { ShadowErrorData }
  CODE                               id-errcode-shadowError }

ShadowErrorData ::= SEQUENCE {
  problem       ShadowProblem,
  lastUpdate    Time OPTIONAL,
```