INTERNATIONAL STANDARD

ISO/IEC 10031-1

First edition
1991-08-01

Information technology — Text and office systems — Distributed-office-applications model —

Part 1: General model

Technologies de l'information — Bureautique — Modèle d'application pour bureau distribué —

Partie 1: Modèle général

Reference number
ISO/IEC 10031-1:1991(E)

Contents

Page

Annexes

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) together form a system for worldwide standardization as a whole. 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. Draft International Standards adopted by the joint technical committee are circulated to national bodies for approval before their acceptance as International Standards. They are approved in accordance with procedures requiring at least 75 % approval by the national bodies voting.

International Standard ISO/IEC 10031-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 10031 consists of the following parts, under the general title: *Information technology – Text and office systems – Distributed-office-applications model –*

– *Part 1: General model*

– *Part 2: Distinguished-object-reference and associated procedures.*

Annexes A to K of this part of ISO/IEC 10031 are for information only.

Introduction

Standards for Open Systems Interconnection permit the functional components of an application to be distributed over a network. Some applications may be distributed in order to reduce costs, e.g. in the case of an office system connected by a high speed Local Area Network (LAN), where some expensive resources may be shared. Other applications may be distributed for administrative or functional reasons, e.g. in the case of a world wide electronic mail system. These are examples of distributed applications whose design differs from the traditional "single host" approach. In general, the increasing range of technological options permits a number of design approaches with quite a different distribution of costs and computational load on the various system elements, such as the desk top devices that populate a modern office and the "back room" devices with which these users are networked.

The purpose of ISO/IEC 10031 is to establish a general framework for distributed-office-applications based on the Remote Operations Service Element.

ISO/IEC 10031 is one of a series, relating to Open Systems Interconnection. Open Systems Interconnection standards are intended to facilitate homogeneous interconnection between heterogeneous information processing systems. ISO/IEC 10031 is within the framework for the coordination of standards for Open Systems Interconnection defined by ISO 7498.

A particular emphasis of ISO/IEC 10031 is to specify the homogeneous externally visible and verifiable characteristics needed for interconnection compatibility, while avoiding unnecessary constraints upon and changes to the heterogeneous internal design and implementation of the information processing systems to be interconnected. It does this by specifying a general model and a set of design principles that will ensure that different distributed-office-applications will function together in a coherent manner.

ISO/IEC 10031 defines the necessary common framework that will enable distributed-office-applications to be developed. Additionally, it provides the concepts and principles applicable to distributed-office-applications that will enable:

a) the modular, simple and expandable development of related products;
b) their implementation from the services of different vendors or service providers;
c) their interworking;
d) the optimization of development costs.

In the interest of effective standardization, ISO/IEC 10031 is oriented towards well understood needs. It is capable of modular extensions to cover future developments in technology and needs.

Although mainly intended for distributed-office-applications, the contents of ISO/IEC 10031 may be used in other information processing environments.

Information technology – Text and office systems – Distributed-office-applications model – Part 1: General model

1 Scope

ISO/IEC 10031 provides a framework for the purpose of the development of protocol standards for distributed-office-applications (DOAs). It applies to applications distributed over significant physical distances as well as "closely-coupled" office systems.

ISO/IEC 10031 describes a model. Distributed-office-applications to be standardized shall use the principles specified in it.

ISO/IEC 10031 provides guidelines for the design of protocols which allow access to the various applications and interactions between applications. The protocols for distributed applications are within the OSI Application Layer and conform to the Remote Operations defined in ISO/IEC 9072.

ISO/IEC 10031 embraces the intent that elements of a system conforming to some parts of it can be implemented from devices supplied by different vendors and by different service providers.

ISO/IEC 10031 does not define human-machine interfaces used with the distributed-applications. Also, it does not define the interface between the software that interacts directly with the user and the software of specific applications.

The content of ISO/IEC 10031 is comprised of two parts.

This part of ISO/IEC 10031 describes the general model of distributed-office-applications and is divided into two sections:

a) model ;
b) guidelines for the design of DOA protocols.

ISO/IEC 10031-2 describes the Distinguished-object-reference and associated procedures that may be used by all DOAs.

No requirement is made for conformance to this part of ISO/IEC 10031. Other parts of ISO/IEC 10031 may specify conformance for systems implementing procedures of those other parts.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10031. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 10031 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7498: 1984, *Information processing systems - Open Systems Interconnection - Basic Reference Model.*

ISO 7498-2: 1989,	*Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2 : Security architecture.*
ISO 7498-3: 1989,	*Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 3 : Naming and addressing.*
ISO 8649: 1988,	*Information processing systems - Open Systems Interconnection - Service definition for Association Control Service Element.*
ISO 8822:1988,	*Information processing systems - Open Systems Interconnection - Connection oriented presentation service definition.*
ISO/IEC 8824: 1990,	*Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).*
ISO/IEC 9066-1: 1989,	*Information processing systems - Text communication - Reliable Transfer - Part 1: Model and service definition.*
ISO/IEC 9066-2: 1989,	*Information processing systems - Text communication - Reliable Transfer - Part 2: Protocol specification.*
ISO/IEC 9072-1: 1989,	*Information processing systems - Text communication - Remote Operations - Part 1: Model, notation and service definition.*
ISO/IEC 9072-2: 1989,	*Information processing systems - Text communication - Remote Operations - Part 2: Protocol specification.*
ISO/IEC 9594-2: 1990,	*Information processing systems - Open Systems Interconnection - The Directory - Part 2: Models.*
ISO/IEC 9594-3: 1990,	*Information processing systems - Open Systems Interconnection - The Directory - Part 3: Abstract service definition.*
ISO 9735: 1988,	*Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax rules.*
ISO/IEC 10021-2: 1990,	*Information processing systems - Text communication - Message-Oriented Text Interchange Systems (MOTIS) - Part 2: Overall architecture.*
ISO/IEC 10021-3: 1990,	*Information processing systems - Text communication - Message-Oriented Text Interchange Systems (MOTIS) - Part 3: Abstract Service Definition conventions.*
ISO/IEC 10021-5: 1990,	*Information processing systems - Text communication - Message-Oriented Text Interchange Systems (MOTIS) - Part 5: Message Store: Abstract service definition.*

3 Definitions

3.1 OSI basic reference model definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO 7498:

a) Application Layer;
b) application-entity;
c) application-service-element;
d) Presentation Layer;
e) presentation-connection;
f) protocol;

g) service definition.

3.2 OSI basic reference model Security part definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO 7498-2:

a) authentication;
b) authorization;
c) credentials;
d) security policy.

3.3 Association control service element (ACSE) definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO 8649:

a) application context;
b) Association Control Service Element.

3.4 Presentation service definition

This part of ISO/IEC 10031 makes use of the following term defined in ISO 8822:

a) abstract syntax.

3.5 Abstract syntax notation definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO 8824:

a) ASN.1;
b) external type;
c) Generalized Time;
d) macro;
e) object identifier;
f) UTC Time.

3.6 Reliable transfer service element (RTSE) definition

This part of ISO/IEC 10031 makes use of the following term defined in ISO/IEC 9066:

a) Reliable Transfer Service Element.

3.7 Remote operations service element (ROSE) definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO/IEC 9072:

a) argument;
b) bind-operation;
c) invoke;
d) operation;
e) perform;
f) Remote Operations;
g) Remote Operations Service Element;
h) result;
i) unbind-operation.

3.8 Directory definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO/IEC 9594:

a) attribute;
b) attribute macro;
c) attribute type;
d) attribute value;
e) filter.

3.9 EDIFACT definition

This part of ISO/IEC 10031 makes use of the following term defined in ISO 9735:

a) EDIFACT.

3.10 Message oriented text interchange systems (MOTIS) definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO/IEC 10021-2:

a) body part;
b) IP-message;
c) message.

3.11 Abstract service definition conventions definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO/IEC 10021-3:

a) abstract model;
b) abstract operations;
c) abstract service;
d) abstract service macro;
e) asymmetric;
f) port;
g) refinement;
h) symmetric.

3.12 Distributed-office-applications model (DOAM) definitions

For the purposes of this part of ISO/IEC 10031, the following definitions apply:

3.12.1 accessee: An x-server which can assign distinguished-object-references (DORs) to objects which it manages upon requests from x-clients, and which can perform operations that designate objects by DORs which it assigned.

3.12.2 accessor: An x-server which can perform operations that designate objects by DORs, by accessing Accessees with the DORs.

3.12.3 control-attributes: Attributes, associated with a security-object that, when matched against the privilege-attributes of a security-subject, are used to grant or deny access to the security-object.

3.12.4 control-attribute-package: A collection of control-attributes.

3.12.5 consume-operation: An operation invoked by an x-client to an Accessor that designates objects by DORs.

3.12.6 data-object: An object that represents data.

3.12.7 data-object-value: A value derived from a data object in accordance with a set of rules, or in the absence of such rules, the value of the entire object.

3.12.8 direct-value-access: Data-object access by value, instead of a reference.

3.12.9 direct-value-transfer: Direct transfer of data-object-value, instead of transfer of a reference.

3.12.10 distinguished-object-reference: A unique reference to a real object in a DOA environment.

3.12.11 distributed-office-application: A set of information processing resources distributed over one or more open systems which provides a well defined set of functionality to (human) users, to assist a given office task.

3.12.12 document: A structured amount of information intended for either direct or indirect human perception that can be interchanged, stored, retrieved and processed by means of office applications.

3.12.13 initiator: An x-client which invokes operations requesting DORs instead of data object value to an Accessee, and which invokes operations that designate objects by DORs to an Accessor.

3.12.14 office-data-object: An object that represents office-information.

3.12.15 office-information: Data used in the office environment.

3.12.16 privilege-attributes: Attributes, associated with a security-subject that, when matched against control-attributes of a security-object, are used to grant or deny access to that security-object.

3.12.17 privilege-attribute-certificate: A certificate using privilege attributes.

3.12.18 produce-operation: An operation invoked by an x-client to an Accessee that requests an DOR instead of data object value.

3.12.19 qualified-attributes: Attributes that have qualification for their usage.

3.12.20 referenced-object-access: Access to objects by means of reference.

3.12.21 ROA-operation: An operation invoked by an Accessor to an Accessee.

3.12.22 security-attributes: A general term covering both privilege-attributes and control-attributes. The use of security-attributes is defined by a security policy.

3.12.23 security-object: An entity in a passive role to which access is granted or denied according to an authorization-policy.

3.12.24 security-subject: An entity in an active role that is granted or denied access to security-objects according to an authorization-policy.

3.12.25 user-application process: An application-process that contains an OA-user and one or more clients of distributed(-office)-applications (e.g. x-client, y-client, etc.).

3.12.26 x- : Generic placeholders for specific application names.

3.12.27 x-access: The definition of the functionality of an x-application, as visible between a x-client and a x-server.

3.12.28 x-access-abstract-service: The abstract service between an x-client and an x-server.

3.12.29 x-access-protocol: The protocol used between an x-client and an x-server.

3.12.30 x-application: A distributed(-office)-application of a certain kind, e.g. an Electronic Mail Application or a Filing and Retrieval Application.

3.12.31 x-application-system: The collection of x-clients and x-server-system that together provide x-user the functionality of x-application.

3.12.32 x-client: That part of an x-application which is part of an application-process which contains an x-user.

3.12.33 x-server: That part of an x-application which is part of a x-server-application-process and that provides the functionality specified by an x-access-abstract-service-definition.

3.12.34 x-server-system: The collection of one or several x-servers.

3.12.35 x-system-abstract-service: The abstract service between x-servers.

3.12.36 x-system protocol: Protocol used between x-servers.

3.12.37 x-user: Part of an application-process in the role assumed when using an x-application.

4 Abbreviations

ACSE	Association Control Service Element
ASN.1	Abstract Syntax Notation One
CAP	Control Attribute Package
DOA	Distributed Office Applications
DOAM	Distributed Office Applications Model
DOR	Distinguished Object Reference
EDIFACT	Electronic Data Interchange For Administration, Commerce, and Transport
OSI	Open Systems Interconnection
PAC	Privilege Attribute Certificate
QoS	Quality of Service
ROA	Referenced Object Access
ROSE	Remote Operations Service Element
RTSE	Reliable Transfer Service Element
UTC	Coordinated Universal Time

5 Model

NOTE – For tutorial background information about the concept used in this clause, see annex D.

5.1 DOA Abstract Model

5.1.1 Abstract Model for Access

Distributed-office-applications shall be developed aligning to the client-server abstract model shown in figure 1, using abstract service definition conventions defined by ISO/IEC 10021-3.

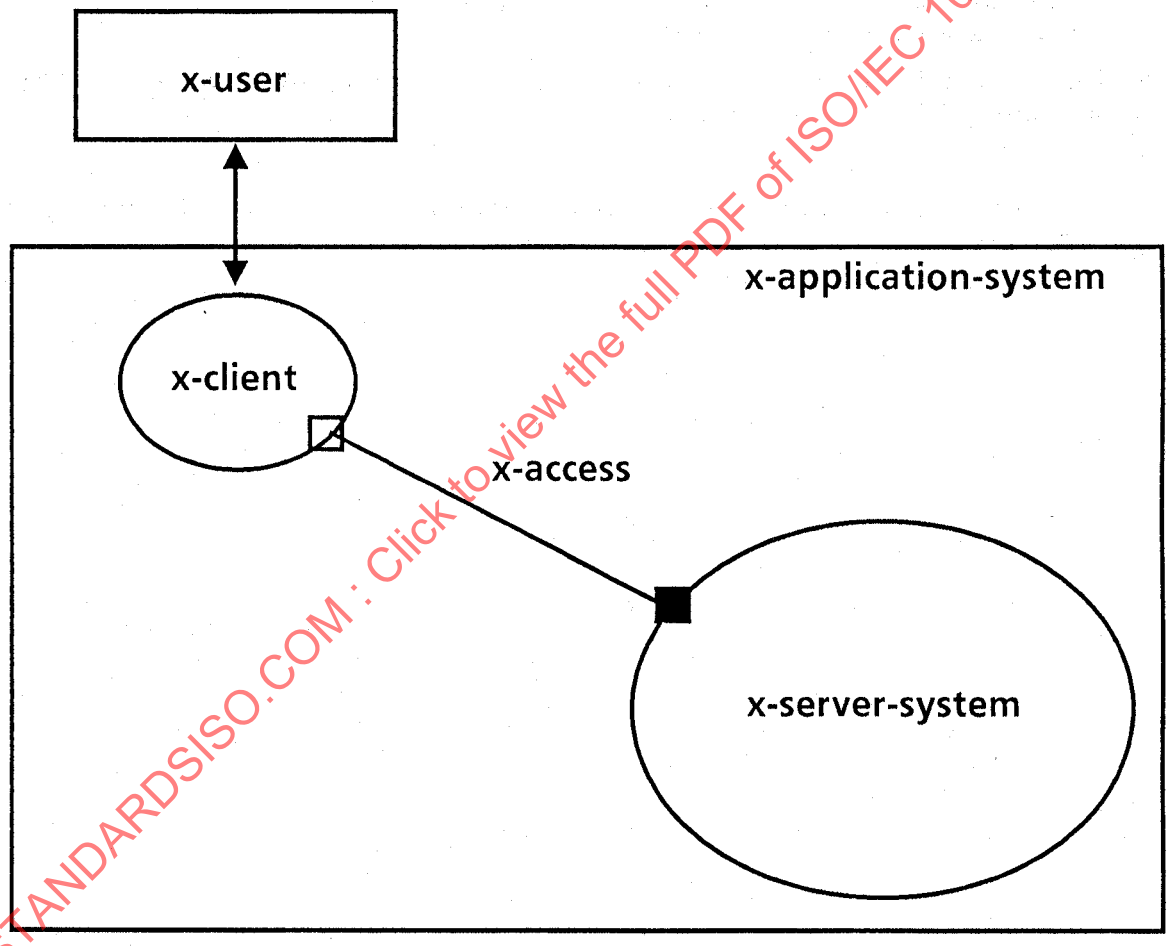


Figure 1 – Distributed-office-applications Abstract Model for Access

In figure 1, the x-user is a user of an x-application, which is provided by the x-application-system. The x-user interacts with x-client to use the service provided by the x-application. The x-client accesses to the x-server via the x-access. The x-server-system may be divided and distributed to more than one x-servers. Details of the internal structure of x-server-system is defined in 5.1.2.

One or more ports may be defined between the x-client and the x-server. For every port, port type shall be asymmetric.

NOTE 1 – Services provided by symmetric ports for access are outside the scope of this part of ISO/IEC 10031.

Information exchanged between the x-client and the x-server-system shall be Office-information. Office-information is data used in the office environment, for example:

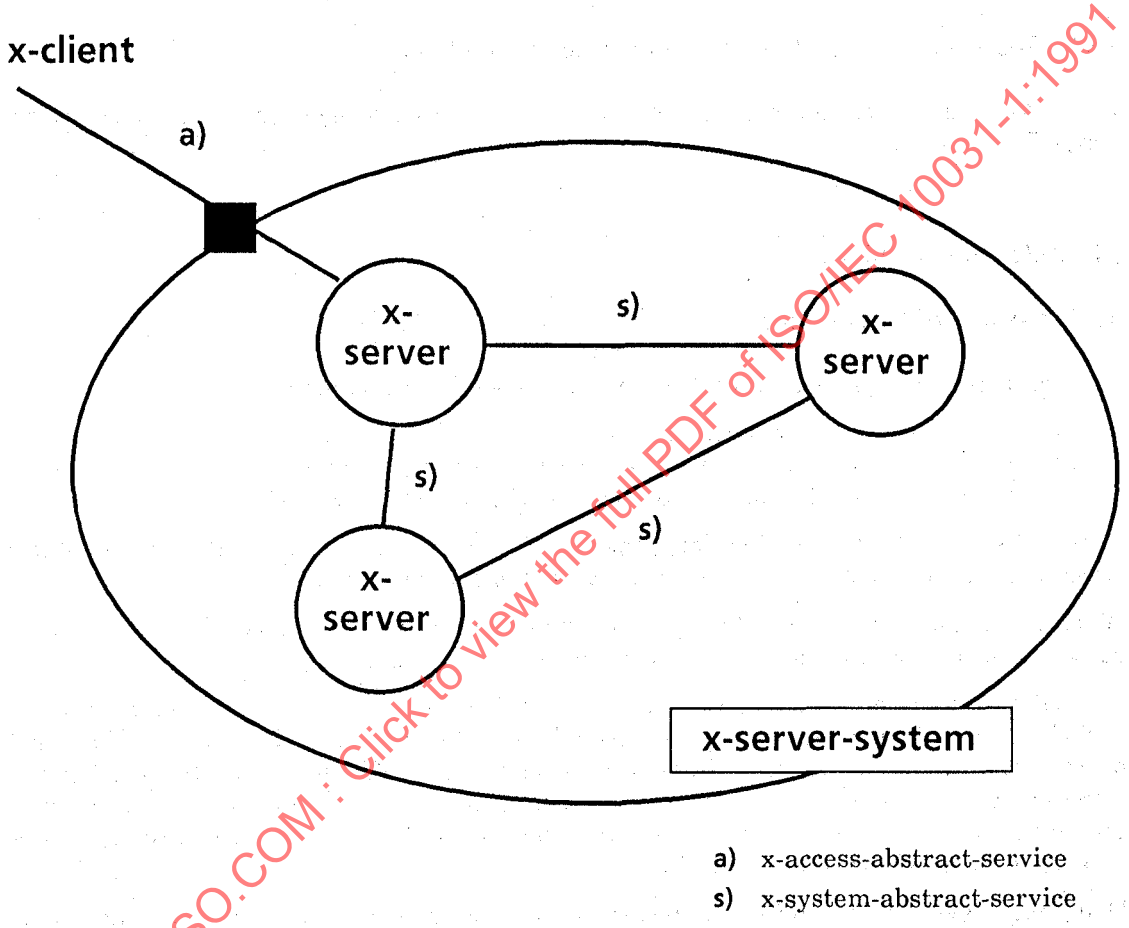
a) documents;
b) messages;
c) EDIFACT data;
d) attributes of documents;
e) time;
f) information relating to messages;
g) information to file documents;
h) information to print documents (including fonts);
i) management information for servers.

This information is viewed as a collection of office data objects, which can be accessed and manipulated individually or in groups.

NOTE 2 – Exchange of information other than Office-information, is, and will be, defined by other International Standards.

5.1.2 Abstract Model for Server-Systems

The x-server-system in figure 1 may be refined to distribute the x-server-system by defining an abstract service between servers, as ISO/IEC 10021-3 suggests. Figure 2 shows a refinement of an x-server-system.



a) x-access-abstract-service
s) x-system-abstract-service

Figure 2 – Refinement of x-server-system

In figure 2, an x-client accesses an x-server-system via the x-access-abstract-service (**a**)). In the x-server-system, an x-server responds to the access. The x-server may interact with other x-servers via the x-system-abstract-service (**s**)) to perform the service requested by the x-client.

An x-server-system may comprise different types of x-servers.

One or more ports may be defined between x-servers. Any type of port may be used.

5.2 Realization of DOA Abstract Model

5.2.1 Realization of Abstract Model for Access

To realize the Abstract Model for access, ROSE defined by ISO/IEC 9072 and its OSI mapping shall be used. The Layer Model is shown in figure 3. Further information about how x-clients and x-servers are identified, see 6.4.4.

5.2.2 Realization of Abstract Model for server-systems

There is no restriction to realize the Abstract model for server-systems. Examples are shown in annex D.

5.3 Referenced-object-access

5.3.1 Classes of Data Access

The access of data object values conceptually involves three parties:

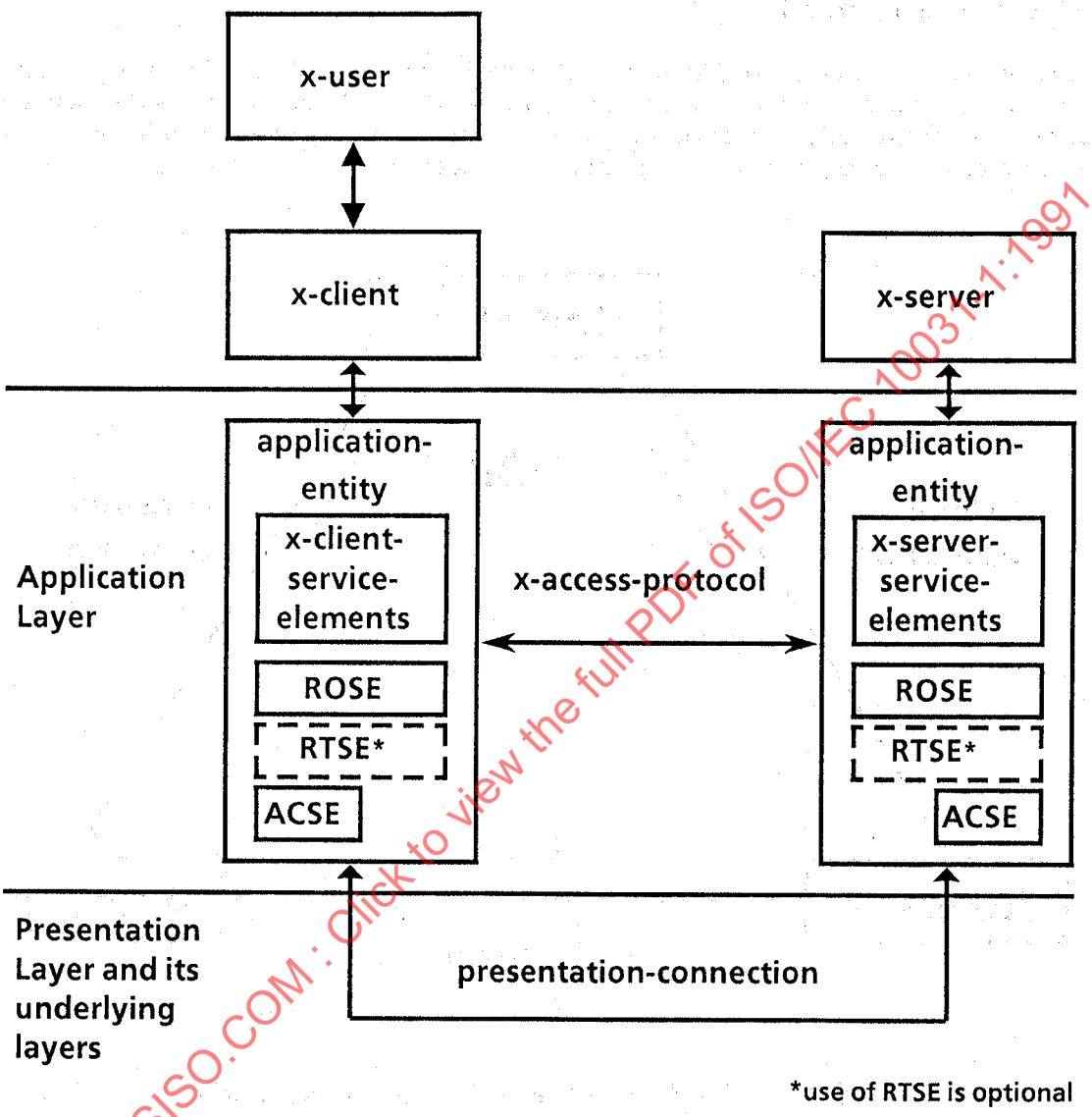
a) an Initiator, which requests the access,
b) an Accessee, which stores and produces the data object value,
c) an Accessor, which consumes or modifies the data object value.

Within distributed-office-applications, there will be applications that will act as an Accessee or Accessor for data objects, for example files, documents or body parts.

When the Initiator is co-located with either the Accessor or the Accessee, data access takes place as part of the access request. This is known as *direct value* access.

In cases where the Initiator is separated from both Accessor and Accessee either physically or by a period of time, using direct value access may involve two data transfers (a "read" and a "write" operation). Alternatively, to use network capacity more efficiently, the Initiator may ask an Accessee to return a *reference* to the data object, rather than its actual value. This reference can then given by the Initiator to the Accessor, which can contact the Accessee directly to access the data object value with a single transfer.

Using high-level programming languages as an analogy, one can consider that the argument or result is passed "by value" when using direct access, and "by name" when using referenced data access.



NOTES

1 This figure is an example and does not restrict mappings.

2 An x-client and an x-server may have several application entities and vice versa.

3 An application entity may serve different server types.

Figure 3 – Layer Model for Realization of DOA Abstract Model for Access

5.3.2 Functional Model for Referenced-object-access

5.3.2.1 Functional Model

The referenced-object-access (ROA) functional model is applicable when an Initiator, Accessee and Accessor are separated either spatially or temporally. Figure 4 illustrates this functional model. For example, the Initiator, Accessee and Accessor may be running on three different systems, or the Initiator system may at some later point in time act as the Accessee or Accessor.

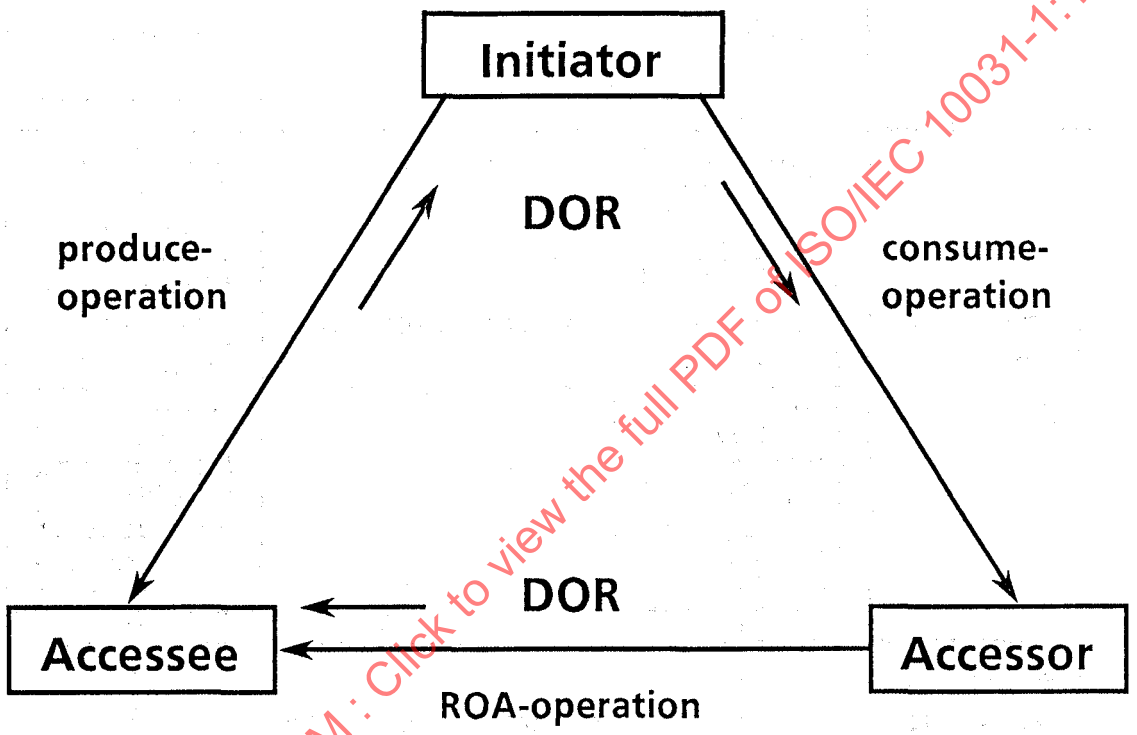


Figure 4 – Referenced-object-access model

In ROA, a reference to the data object value (known as a Distinguished-object-reference, or DOR) is returned to the Initiator in the result of a produce-operation and passed to the Accessor in the argument of a consume-operation. The Accessor then invokes an ROA-operation. In the case of a write operation, a new value, or instructions for modifying the data value, can be passed with the access request. In the case of a read operation, the actual value of the referenced data object is returned in the result of the access operation.

The value accessed when an ROA-operation is performed is related to the data known to the Accessee by application-specific rules that are associated with the DOR during the produce-operation. For example, a DOR could be defined to reference the first Body part of a particular MOTIS IP-message.

ROA-operations are not necessarily constrained to access a fixed or permanent data object.

In figure 4, a DOR is generated by the Accessee in response to a produce-operation invoked by the Initiator. The DOR is offered to the Accessor by the Initiator as a parameter in an consume-operation. The Accessor can then use the DOR to interact with the Accessee.

5.3.2.2 Produce-Operations

In some data operations, the Initiator uses an application-specific protocol to select a data object value (the complete data object, or some subset or derivative of the data object) from an Accessee. This class of operation is termed "produce-operation".

In direct value transfer, the Accessee returns the data object value to the Initiator, and the Initiator takes the role of the Accessor. In the indirect case, on the other hand, the Initiator requests a reference to the data object, rather than the data object value, and the Accessee returns the DOR to the Initiator. DORs identify their associated data value uniquely.

Where DORs are supported by a produce protocol element, a parameter in the Invoke is required to specify whether a direct data value or a DOR is to be returned. The result will accordingly contain a data value or a DOR.

5.3.2.3 Consume-Operations

The Initiator can also use an application-specific protocol to cause a data object value to be accessed by an Accessor. This class of operation is termed "consume-operation". In direct value transfer the Initiator also acts as the Accessee and provides the data object value in the protocol. In the indirect case, the Initiator provides a DOR (previously obtained from an Accessee) to the Accessor. The Accessor uses this DOR to perform an access operation on the Accessee to read or write the referenced data object value.

Where DORs are supported by a consume protocol element, the supplied data may be either a data object value or a DOR. The result will have identical semantics in both cases, but if a DOR is used, the Accessor may need to wait for the result of the ROA-operation before returning the result of the consume operation.

5.3.2.4 ROA-operations

The distributed-office-applications model defines a special class of protocols which always use DORs to interact with Accessee Application-objects, and which provide a generalized set of operations. This class is called "Referenced-object-access protocols (ROA-protocols)".

5.3.2.5 Implications of supporting DORs

DORs require additional functionality to be built into the Accessee and Accessor:

a) The Accessee needs to be able to provide a DOR rather than a data object value in response to a produce-operation;
b) The Accessor needs to be able to accept a DOR rather than a data object value in a consume operation;
c) The Accessor needs to be able to invoke an access-operation;
d) The Accessee needs to be able to perform an access-operation.

Within application-specific protocols, standards may chose whether:

a) to allow DORs to be used in any protocol element where a reference to a data object is supplied or returned;
b) to place additional restrictions on where DORs are allowed;
c) to define special protocol elements to handle the ROA-produce and ROA-consume operations.

The first of these option is strongly preferred, and shall be used whenever possible.

If either Accessee or Accessor in a particular proposed access does not support DOR, then the Initiator has no choice but to execute two consecutive direct-value transfers. In this case, a produce-operation invoked by the Initiator returns a data object value from the Accessee to the Initiator, and the Initiator transfers that data object value to the Accessor in the argument of a consume-operation. (This description applies to a read operation. In the case of a write operation, the data will flow in the opposite direction.)

5.3.2.6 Quality of Service

The value of some data objects will change with time, or the object may be deleted. Individual protocols may chose whether DORs:

a) refer to the value of the data object when the DOR was generated;
b) refer to the current value of the data object;
c) become undefined if the object is updated.

To assist in the control of references to dynamically changing objects, DOR may include a Quality-of-service (QoS) indication. The QoS describes the expected or required scope for the validity of both the DOR and the value of the associated data. X-access protocols may need to support protocol elements to update the QoS.

5.3.2.7 Structure of DORs

Details of the structure of the DOR and its associated procedures are defined in ISO/IEC 10031-2.

6 Guidelines for the design of protocols

6.1 Introduction

This clause shows the guidelines for the design of protocols to be adhered to by all distributed-office-applications standards.

6.2 Office-information

The main purpose of distributed-office-applications is to interchange, store, retrieve and process office-information.

To maintain the variety of existing or future concepts and types of office-information, the abstract syntax and the semantics of the office-data-object may typically be transparent to the protocols of distributed-office-applications. In this case, the office-data-object shall appear as an ASN.1 external type in the "direct-reference" variant (i.e. no presentation layer negotiation of encoding rules) in the abstract syntax of DOA protocols. The "direct-reference OBJECT IDENTIFIER" value of the external type references both the abstract syntax and the encoding of the object. This value shall be used in attributes identifying the object type.

6.3 Object model and remote operations

6.3.1 Use of remote operations

The Remote Operations defined in ISO/IEC 9072 provide a notation and a protocol specification for bind-operations, unbind-operations and operations (called type-operations in the object model). A standard set and naming guidelines for operations are described in the following subclauses.

All access protocols for distributed-office-applications shall conform to the Remote Operations as specified in ISO/IEC 9072. In more detail the access protocols shall use the notation and concepts of that International Standard and shall allow any mappings defined in ISO/IEC 9072-1 clause 11 . Annex J gives a short introduction to these concepts in the context of an access protocol taking into consideration the application rules of 6.4.

For system protocols, it is encouraged to also use Remote Operations wherever possible.

6.3.2 Use of the abstract service technique for the x-service-definition

The abstract service technique is based on a number of ASN.1 macros which are used to describe the function and parameters of services. This description technique for services is closely related to the way Remote Operations are formally defined. The technique guarantees full consistency between service definitions and protocol specifications. It also saves duplication of work and documentation through the possibility to import definitions made for services into the formal protocol. It is also very easy to import definitions made in one DOA to another DOA without having to redefine or redocument them. Standards for MHS and Directory are already using this technique. All future DOAs shall use the same technique for the documentation of services.

Abstract service macros are defined in ISO/IEC 10021-3.

6.4 Application rules

The following rules have been established in order to simplify the management of shared resources among a number of applications.

6.4.1 Concurrency and resource sharing

6.4.1.1 Concurrency

There are established techniques in centralized systems to control concurrent access and to preserve the integrity of data. For distributed systems there are no economic general solutions to the general case of distributed data.

Applications should avoid the general case. Until a stronger requirement arises and a solution is worked out for a specific application, the guideline is to use a managed weak consistency - that is:

a) allow inconsistent data;

b) have one master copy for each item of data, and have one specific server responsible for updating it;

c) have one sequence of propagating changes to copies of that data item and changes to related data items;

d) minimize relationships between data items in different servers;

e) provide administrative controls to regulate how long it takes for a change to propagate;

f) design applications to be tolerant of or resilient to out of date data.

With this guideline, concurrency controls can be limited within a single x-server, or at most within an x-system. The impact on protocols is limited to the impact of resource sharing.

6.4.1.2 Resource sharing

Shared resources within a server are managed by the server (which in turn relies on the underlying operating system of the node).

The impact on protocols is limited to managing the consequences of a server not being able to respond to an interaction for a time. This can be manifested by a refusal to accept an interaction, a response indicating a delay or a deferred response. The entity acting as an x-user may impose time-out disciplines.

If required, an x-system may provide for management of resources shared between its x-servers. This will require to be expressed in the x-system-protocol, but will not impact the x-access-protocol except as described above.

6.4.2 Network transparency

To insulate users from the details of environmental network configuration, servers and clients shall be referred by name rather than by the Presentation Address. The Directory may be used to provide this translation.

6.4.3 Common definition of time

All protocols in the distributed-office-application environment shall express the time using the data type "GeneralizedTime" as defined by ISO 8824.

Time :: = GeneralizedTime

NOTE – ISO/IEC 9594 and ISO/IEC 10021 currently use **"UTCTime"** instead of **"GeneralizedTime"**. Use of **GeneralizedTime** in these standards, keeping backward compatibility, is planned.

6.4.4 Common definition of identifiers

All objects defined in DOA-standards shall have at least one globally unique name. A common perception of names and a common definition of identifiers are defined in ISO 7498-3, ISO 8824, and ISO/IEC 9594. They are introduced in annex E.

The ASN.1 encoding of names used by applications adhering to this Model is as defined in ISO/IEC 9594.

6.4.5 Use of attributes and filters

Many objects in the context of distributed-office-applications (e.g. represented in an information base) are characterized by attributes. An attribute consists of an attribute type, which identifies the class of information given by that attribute and the corresponding attribute value(s).

The attribute concept, a notation supporting the definition of attributes, and the abstract syntax of attributes are defined in ISO/IEC 9594-2. A subset is defined in ISO/IEC 10021-5. Standards for distributed-office-applications shall use attributes if appropriate and shall refer to ISO/IEC 9594-2.

If objects represented by entries in an information base are characterized by attributes, information retrieval (e.g. entry selection) may require filters. A filter applies a test that is either satisfied or not (e.g. by a particular entry). The filter is expressed in terms of assertions about the presence or value of certain attributes (e.g. of an entry).

The semantics and abstract syntax of filters are defined in ISO/IEC 9594-3, and a subset is defined in ISO/IEC 10021-5. Standards for distributed-office-applications shall use filters if appropriate and shall refer to ISO/IEC 9594-3 .

Attribute Types defined for one application may be reused by another application as long as the definitions and the semantics are the same. The OBJECT IDENTIFIER defined in ISO 8824 is used as a tool to achieve this.

The Attribute Macro is defined in ISO/IEC 9594-2.

The technique of "qualified attributes" may be considered for use in distributed-office-applications. This allows for example a client to flag whether a particular attribute is compulsory in the sense that the corresponding server has to understand the semantics of the attribute and know how to respond, or if the server can ignore the attribute or substitute a default.

NOTE – For examples of the usage of this technique, see ISO/IEC 10175-1:1992[1], *Information technology – Text and office systems – Document printing application – Part 1: Abstract service definition and procedures.*

1 To be published.

6.4.6 Referenced-object-access

Data object values and DOR in access protocols may typically appear as ASN.1 external types.

6.4.7 Application-service-elements and application context.

Application-service-elements implementing the functions required for several access protocols may be combined into a single application context. This requires distinct abstract syntaxes for each of these application-service-elements. (See also figure 3.)

6.5 Security rules

6.5.1 Introduction

The following rules have been established in order to simplify the implementation and management of the security aspects of Access Control and Authentication.

The rules enable a wide range of security policies to be used without change to the distributed-office-applications protocols, including security policies that demand individual accountability of persons using the distributed-office-applications.

NOTE - For detailed tutorial introduction to the security concepts, see annex F.

6.5.2 Security subject

The security subject is often the individual accountable for the operations being performed. In some security policies, accountability may be vested in a group of people, or in a user. Distributed-office-applications may control access in terms of the security subject identity, or in terms of capabilities claimed by the security subject (there are techniques to validate such claims). A Privilege Attribute Certificate is a data structure that securely carries the attributes of security subject; the values present in the structure depend upon the security subject and the security policy in force.

Distributed-office-applications shall use a Privilege Attribute Certificate (PAC) to signify the authenticity and privileges of the security subject. Where the security policy in force requires individual accountability, the PAC shall include the necessary identification.

The PAC shall be passed on the BIND, and it shall apply to all subsequent operations in the association until UNBIND or ABORT terminate it. Each individual operation shall allow the client to pass another PAC; this shall supplement the BIND's PAC for that individual operation.

Where an operation causes processing subsequent to the operation, or subsequent to the UNBIND or ABORT, the the PAC(s) in force for the operation shall also apply to the subsequent processing.

6.5.3 Security objects

The security object is the object that is being protected in order that access by security subjects is regulated by the security policy in force. The security policy may require that checks are made on the basis of privileges claimed by the security subjects identity, or on the basis of privileges claimed by the security subject, or some composite of the two approaches. A Control Attribute

Package (CAP) is a data structure that securely carries the access control information; the values present depend upon the security policy in force and the individual object.

Distributed-office-applications shall use a Control Attribute Package (CAP) to convey the access control information of an office-data-object. This can occur when creating an object, modifying an object, or transferring an object with its access control information.

6.5.4 Access Control

The preceding security rules separate out the design of distributed-office-applications protocols from the design of security elements conveyed within the protocols. This facilitates the use of the same protocols in differing security environments.

The operation definitions and data structure models of distributed-office-applications shall permit a wide range of security policies to be used in conjunction with their protocols.

Any security policy assumptions, such as implicit copying of access control attributes from one security object to another, should be qualified by "subject to the constrains of the security policy in force".

6.5.5 Security Errors

Almost any operation may be wholly refused on the grounds of security.

Security refusals shall be reported as errors, taking precedence over other errors (except for protocol errors which prevent the determination of the security subject, the security object and the operation). Security error responses shall not be phrased in such a way, nor accompanied by, nor followed by other error responses that could imply or convey information that the security subject should not be given.

Operations that cause one access to a single security object shall have no effect on the object if that access is refused.

Operations that access several security objects may encounter security refusals on some of those objects. The resulting effects and responses of such operations shall be defined.

6.6 Standard set of operations

This subclause shows the standard set of abstract operations.

Its primary purpose is to harmonize sets of abstract operations and their names for various distributed office applications. This harmonization will reduce the time and effort needed to standardize specific distributed office applications, and will make it easier for implementors of DOA standards to become familiar with and apply operations of a specific application.

The developers of a specific distributed-office-application shall use the operations of this standard set to define the set of operations in their specific distributed-office-application where appropriate.

The following is the standard set of DOA abstract operations:

a) List;
b) Read;
c) Modify;
d) Copy;
e) Move;
f) Search;

g) Create;
h) Delete;
i) Reserve;
j) Notify;
k) Abandon.

An example of the details of those operations are shown in annex K.

Annex A
(informative)
References, definitions, and abbreviations for informative annexes

A.1 Introduction

This annex shows references, definitions and abbreviations specific to the informative annexes of this part of ISO/IEC 10031. References, definitions and abbreviations which are shown in the body of this part of ISO/IEC 10031 are not included.

A.2 References

The following standards contain provisions which, through reference in this text, constitute provisions of the annexes of this part of ISO/IEC 10031. At the time of publication,the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 10031 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7498-4: 1989,	*Information processing systems - Open Systems Interconnection - Basic Reference Model Part 4 : Management framework.*
ISO 8571: 1988,	*Information processing systems - Open Systems Interconnection - File Transfer, Access and Management (FTAM).*
ISO 8613:1989,	*Information processing systems - Text and office systems - Office Document Architecture (ODA) and interchange format.*
ISO / IEC 8879: 1986,	*Information processing - Text and office systems - Standard Generalized Markup Language (SGML)*
ISO/IEC 9545: 1989,	*Information technology - Open Systems Interconnection - Application Layer structure (ALS).*

A.3 Definitions

A.3.1 OSI basic reference model definition

The annexes of this part of ISO/IEC 10031 make use of the following term defined in ISO 7498:

a) application-process.

A.3.2 OSI basic reference model security part definitions

The annexes of this part of ISO/IEC 10031 make use of the following terms defined in ISO 7498-2:

a) access control;

b) access control list;
c) audit;
d) audit trail;
e) authentication information;
f) capability;
g) confidentiality;
h) data integrity;
i) data origin authentication;
j) digital signature;
k) encryption;
l) key;
m) key management;
n) repudiation.

A.3.3 Message oriented text interchange systems (MOTIS) definitions

The annexes of this part of ISO/IEC 10031 make use of the following terms defined in ISO/IEC 10021-2:

a) message transfer;
b) message transfer system;
c) message store;
d) P2;
e) user agent.

A.3.4 Distributed-office-applications model (DOAM) definitions

For the purposes of the annexes of this part of ISO/IEC 10031, the following definitions apply:

A.3.4.1 access-control-policy: A set of rules, part of a security policy, by which human users, or their representatives, are authenticated and by which access by these users to services and security-objects is granted or denied.

A.3.4.2 access-context: The context, in terms of such variables as location, time of day, level of security of the underlying associations, etc, in which an access to a security-object is made.

A.3.4.3 cryptographic key: See key.

A.3.4.4 data-object-format-specification: A data type, in the ASN.1 sense, that is defined independently of x-access-protocols.

A.3.4.5 node: A data processing facility that provides information processing resources as part of a network. A node may support user-application-processes, server-application-processes or a combination of both kinds of processes.

A.3.4.6 OA-user: A part of an application-process which directly interacts with a human user, and which is using one or more office applications on behalf of the human user.

A.3.4.7 security-administrator: An authority (a person or group of people) responsible for implementing the security policy for a security-domain.

A.3.4.8 security-domain: A bounded group of security-objects and security-subjects to which applies a single security policy executed by a single security-administrator.

A.3.4.9 security-facility: Procedures, processes, mechanism or set thereof, that models a security related function.

A.3.4.10 server-application-process: An application-process that implements all or part of the functionality defined by an x-service-definition.

A.3.4.11 user: A human user or an x-user.

A.3.4.12 user-application-process: An application-process that contains an OA-user and one or more clients of distributed(-office)-applications (e.g. x-client, y-client, etc.).

A.3.4.13 x-, y-, z-, ...: Generic placeholders for specific application names.

A.3.4.14 x-application-interface: The interface to an x-application, as visible between an x-user and an x-client.

A.3.4.15 x-service-definition: The definition of the functionality of an x-application, as visible between x-clients and the x-system.

A.4 Abbreviations

ASE	Application-service-element
CCR	Commitment, Concurrency and Recovery
DSSSL	Document Style Semantics and Specification Language
FTAM	File Transfer, Access and Management
MS	Message Store
MOTIS	Message Oriented Text Interchange Systems
OA	Office Applications
ODA	Office (Open) Document Architecture
ODP	Open Distributed Processing
RDA	Remote Database Access
SGML	Standard Generalized Markup Language
SPDL	Standard Page Description Language
TP	Transaction Processing
VTP	Virtual Terminal Protocol

Annex B
(informative)
Relationship to other standards

B.1 Context

This part of ISO/IEC 10031 addresses aspects of an integrated, yet distributed office system supporting professional, technical and administrative users. This part of ISO/IEC 10031 does not address real time nor transaction processing which generally support frontline operational staff, such as point of sale, reservations clerks, cash dispensers, etc.

This part of ISO/IEC 10031 addresses the use of multi-vendor equipment through the use of OSI, and it addresses interworking between organizations. There is provision for security; incorporation of firm detailed guidelines will be done when OSI security standards become clearer and more stable.

B.2 Use of other standards

Protocol development under this model is carried by ROSE over the normal mode OSI Upper Layer protocols. Directory is assumed to be available throughout the distributed office system.

B.3 Synergy with other standards

Message Oriented Text Interchange Systems (MOTIS) and Directory set precedents on a number of aspects common to distributed-office-applications. For historical reasons, MOTIS has aspects which do not fully align with this part of ISO/IEC 10031.

Documents as processable information, printable information, mailed information are the major form of office information. While the protocols to be developed under this model do not depend upon the use of particular encodings of documents there will be synergy between the existing and developing document standards of Office (Open) Document Architecture (ODA), Standard Generalized Markup Language (SGML), Standard Page Description Language (SPDL) and Document Style Semantics and Specification Language (DSSSL).

Remote Database Access (RDA) may be used as an office function and some harmonization with DOAM would be beneficial.

B.4 Coexistence with other standards

B.4.1 Virtual terminal protocol (VTP)

Since VTP is located between applications and the human user, it is outside the scope of this part of ISO/IEC 10031.

B.4.2 File transfer protocols

File Transfer Access and Management (FTAM) are not directly perceived by the majority of office system users who operate in terms of documents and relational-like data access. This part of ISO/IEC 10031 lays down guidelines that are not fulfilled with FTAM. Nevertheless, user applications that wish to use FTAM-like file store models and functionality will also be able to use FTAM as well as office applications.

B.4.3 Transaction processing (TP) and commitment, concurrency and recovery (CCR)

There are no requirements at present to use office applications in the environment of distributed TP. There are no requirements at present for CCR disciplines. This part of ISO/IEC 10031 currently does not therefore require office application protocols to be designed in a style that might fit these forthcoming standards.

B.4.4 Open distributed processing (ODP)

This part of ISO/IEC 10031 does not currently take the ODP Work Item into account. However, harmonization is expected in due course.

Annex C
(informative)
Requirements

C.1 Introduction

Distributed-office-applications are used by an integrated distributed office system. A distributed office system consists of user nodes and server nodes linked by a network. The user nodes access the server nodes via the network, using access protocols. An integrated office system maintains coherent cooperation of the various office applications.

In such an environment, data processing applications, that within a single host act as a single unit, have been split among the different intelligent components of the system. This splitting has led to the need for standardization of inter-relationships between the different parts of an application.

In a distributed office system, simultaneous availability of all resources can not be guaranteed and neither supportive nor productive applications should assume that all parties to a particular distributed process (e.g transmittal of a message) are simultaneously in communication unless demanded by the semantics of that process. This leads to the concept of store and forward communication, for example for message transmittal where neither the originator nor the potential recipient nor intermediary message transfer agents are on-line during the same period of time.

C.2 Functional requirements

Amongst the services that may be required by the user from a distributed office system, and which the framework should address, are:

a) Inter-personal messaging, to communicate with other users;

b) Group communication, to communicate with groups of users;

c) Conversion, to allow the interchange of documents with different syntaxes or character coding;

d) Filing and retrieval of documents, to allow an ordered filing and multi-key retrieval of documents;

e) Input and output of documents to the distributed office system from different physical devices, such as scanners, printers, etc;

f) Directory, to know where are and how to access remote communication elements, applications or users;

g) Authentication, to avoid unauthorized access to the different applications;

h) A time base which is locally accessed, for purposes such as time stamping messages and files throughout the network;

i) Direct access to remote servers (e.g. Videotex) and users (e.g. via Teletex);

j) Indirect communication (store and forward) with remote systems, to transfer non real-time information or to access other networks for example Message Transfer System;

k) Data transfer between different applications or remote servers.

The above list of applications will grow in the future (e.g. data base access). Most of these additional applications are likely to be productive applications because their main purpose is to provide specific facilities to office workers.

The typical usage of these applications needs a high degree of integration. For example, a document may be fetched from a message store server, stored on a document filing and retrieval server, and printed on a printing server.

The operation of some of the above distributed-office-applications (more usually Group Communication, Document Filing and Retrieval, Printing) may need the use of other applications which perform supportive application roles (e.g. Directory, Authentication).

However, applications and their users should not be impacted by the way supportive applications are carried out. In particular, the actual distribution should be as transparent to the user as possible.

Supportive applications are not necessarily visible to the human user, but they do enable secure, reliable and smooth operation of the overall system.

C.3 Design requirements

The design of the protocols should ensure

a) Stability:

the recommended design principles shall be such that distributed-office-application protocol designers can specify highly stable distributed-applications interactions, with comprehensive use of common supportive applications;

b) Modular design:

1) minimizing inter-dependencies between different office applications of a yet open-ended list of productive applications;

2) enabling the required high degree of integration;

c) Common style of protocol design;

1) Homogeneous usage of supportive applications and facilities;

2) Homogeneous usage of the Remote Operations Service Element;

d) Security: of various levels specified by the user;

e) Simplicity: this is a key point from the human user's point of view of distributed-office-applications. This means that the user should not be involved in the way its request is managed by the applications;

This part of ISO/IEC 10031 specifies the principles of interaction between the various applications.

Annex D
(informative)
Basic concepts

D.1 Introduction

Distributed-office-applications are categorized as a number of applications which, from the human user point of view, constitute an integrated office system.

A distributed office system consists of nodes connected through a network. While this system exists for the benefit of human users, any part of an application-process may use the office applications.

A user node is the device with which a human user directly interacts. It provides direct interactive functions.

A server node is a device that manages resources which are shared among many users.

A human user's interactions with a distributed office system may cause a number of activities to be performed on a number of nodes. How these separate activities on one node are represented by processes, tasks, etc, is not the concern of this part of ISO/IEC 10031. The interactions between one activity on one node and one activity on another node are represented in the OSI model by interactions between a pair of application process invocations, one on each node. An application process invocation executes the functionality of an OSI application-process.

For clarity, this subclause describes interactions in terms of application-processes. Each functional entity described in this sub-clause is related to an application-process; each separate activity executing the functions is related to a separate application process invocation.

D.2 Client-server model

This subclause adopts a tutorial approach to explain the distribution of a single x-application, an application of a certain kind.

D.2.1 Non-distributed single application

In a non-distributed single x-application, the x-user and the x-application are co-located. In the special case where the x-user is interacting with a human user and interacting with the x-application on behalf of the human user, the x-user is termed OA-user and the application-process is termed user-application-process (see D.3.1). The x-user interacts with the x-application through an x-application-interface, which is usually proprietary and is not proposed for standardization (see figure D.1).

If an x-application is a candidate for distribution, an x-service-definition is needed as a separation line for potential future distribution.

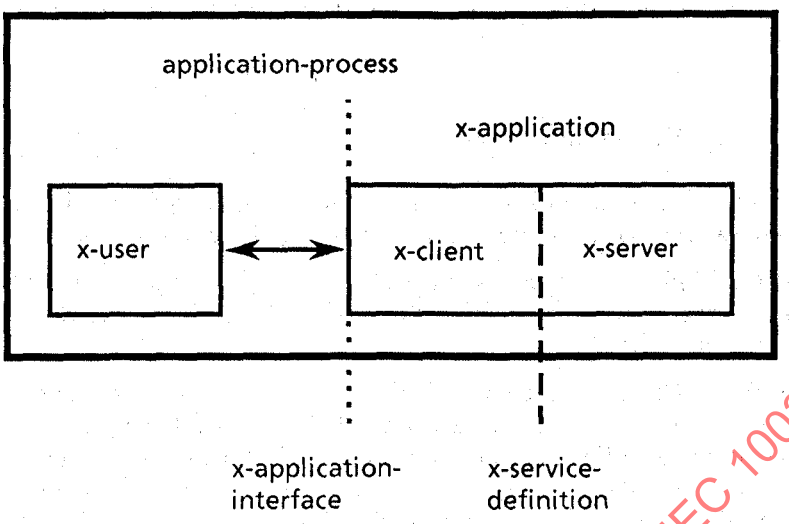


Figure D.1 – Non-distributed Office Application

D.2.2 Distributed single application

The distribution of an x-application should be transparent to the x-user, so that the x-application-interface will not change. The x-client is co-located with the x-user. The x-user and the x-client are together within one application-process.

The x-server is generally remote from the user. The x-server is part of an application-process which is termed a server-application-process.

An x-client and an x-server communicate over the network by means of an x-access-protocol. There may be several independent interactions between an x-client and an x-server.

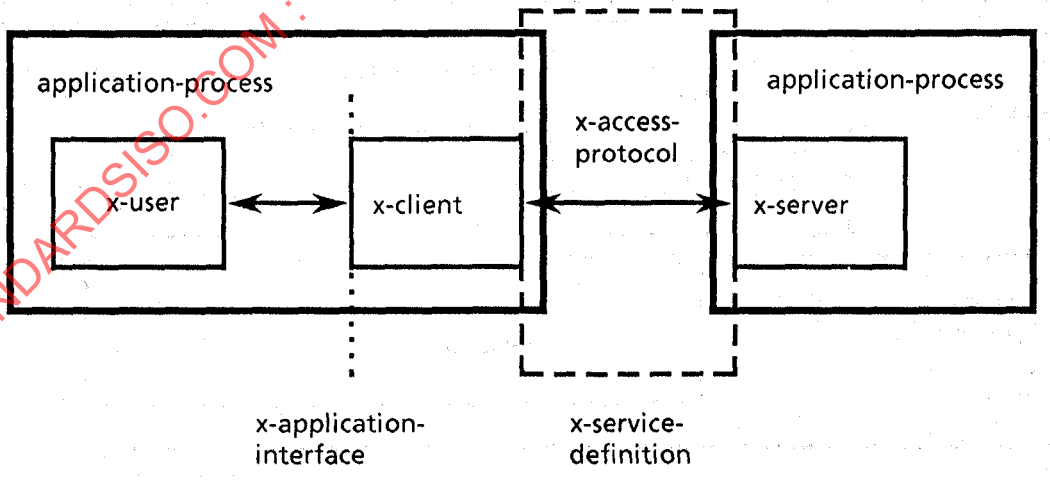


Figure D.2 – Distributed-office-application

The new situation is depicted in figure D.2 which shows the x-service-definition of figure D.1 expanded into a "dashed" box encapsulating the x-access-protocol.

In the case of distribution, the x-service-definition and the x-access-protocol need to be standardized.

D.2.3 Client-server OSI communication

An x-access-protocol is the standard way for an x-client to gain access to its remote x-server. The following model shows how OSI principles are used to specify an x-access-protocol.

According to the OSI Reference Model, a protocol is used between peer entities. In figure D.2, OSI communications occur only within the dashed box. Thus, the OSI communication peer entities are inside, not outside that dashed box.

In compliance with the OSI Reference Model, the x-client and the x-server are considered part of an application processes, and have application-entities associated with them. The application-entities are part of the Application Layer of the OSI Reference Model and contain sets of application-service-elements. The application-service-elements provide the communication functions, in accordance with the service definition, to the x-client and the x-server, and implement the x-access-protocol. In doing so, an application-service-element may use services provided by other application-service-elements in the same application-entity, and by the Presentation Layer of the OSI Reference Model.

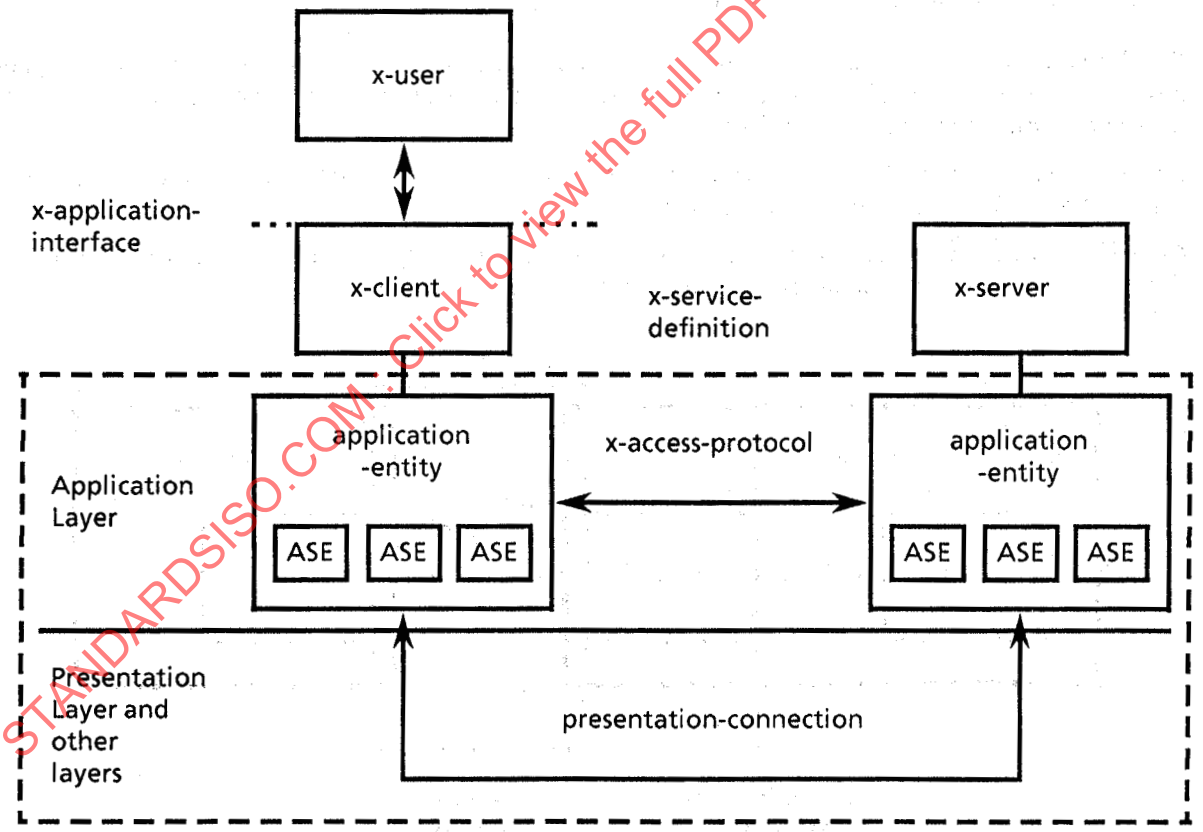


Figure D.3 – Distributed-office-application with OSI Communication

In figure D.3, the dashed box is expanded to show a more detailed model of the OSI communication between the x-client and the x-server.

In detail, interactions take place between application entity invocations. Discrete sets of interactions between the same pair of application process invocations, if required, are performed between discrete pairs of application entity invocations.

However, for most practical purposes, it is not necessary to refer to the above detailed structure. Instead, the model of x-client/x-server communication by means of an x-access-protocol is generally sufficient for discussing the structure of distributed-office-applications.

Additional details about the client/server communication are specified in D.4.

D.2.4 Object Model of a distributed-office-application

The interworking of the application-processes of a distributed-office-application requires a shared conceptual schema describing the shared universe of discourse.

The typical universe of discourse is composed of objects and relationships between them and may provide a classification of objects.

The client/server model is considered as an application oriented tool to develop conceptual schema. The object model has broader view and is more abstract. The components of a distributed-office-application (e.g., mailbox, filing cabinet, etc.) are all addressed as objects.

The tools which were developed within the object model are used here for the specification of the conceptual schema. The conceptual schema is the basis for the service-definition.

It is worth noting that another kind of object type is used in the context of distributed-office-applications. These are data objects (e.g. ASN.1 data types, message contents, Interpersonal Message body parts, private document formats). The abstract syntax of these data objects is defined independently of x-access-protocols and x-system-protocols.

D.3 Functional Model

D.3.1 Multiple applications in an integrated system

An integrated office system is formed by a set of office applications (e.g. Message Transfer, Document Filing and Retrieval and Printing). The integration of the office applications is performed by the OA-user. The OA-user interacts with the human user and, on behalf of the human user, with the set of office applications. Interactions between the OA-user and an x-server are performed via the x-client. Interaction between clients is performed via the OA-user. This is depicted in figure D.4.

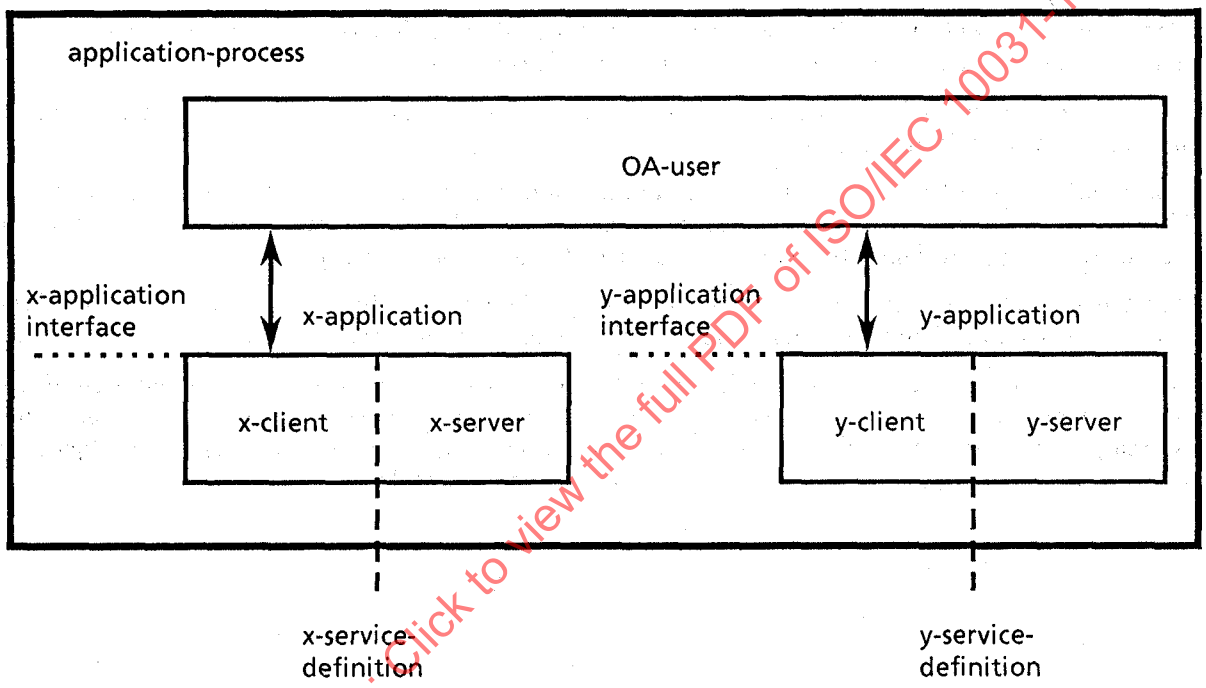


Figure D.4 – Multiple Non-distributed Office Applications

D.3.2 Multiple distributed-office-applications

The OA-user and the clients are in one application-process which is termed the user-application-process (see figure D.5). Several user-application-processes may coexists on one user node, but these are kept separate within this part of ISO/IEC 10031.

For some applications (e.g. Directory, Message Store) the representation of each OA-user accessing a server is called a User Agent. This User Agent contains the clients and the OA-user.

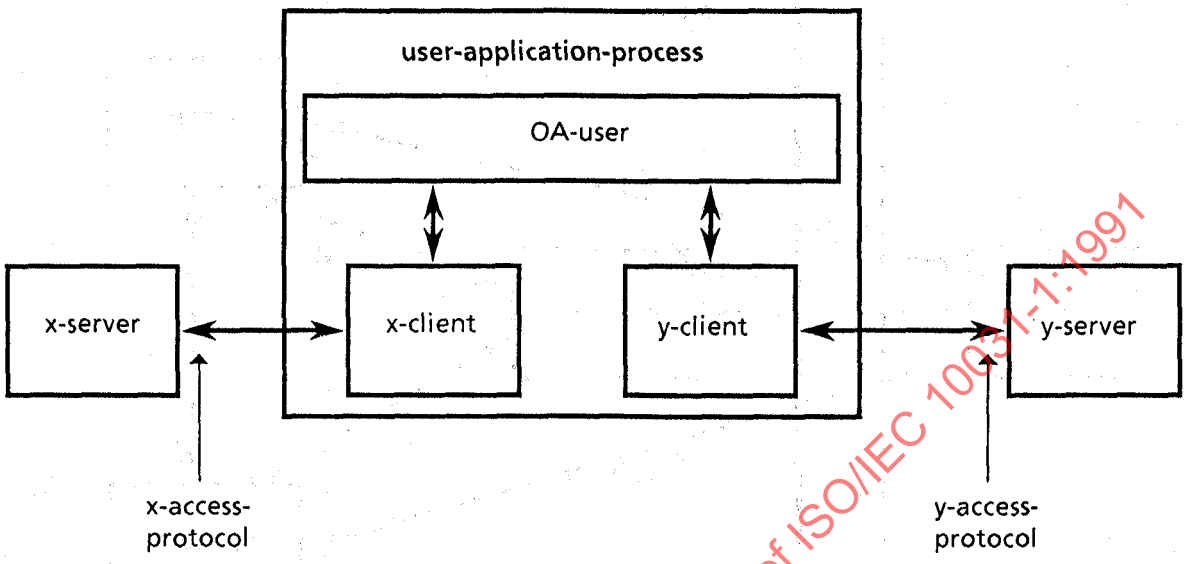


Figure D.5 – Multiple Distributed-office-applications

D.3.3 Organization of servers

There can be a second distribution step, distributing the functionality of the server part of an x-application to several x-servers on several nodes. The set of x-servers is termed an x-system. Each x-server is functionally equivalent in that it supports the same x-access-protocol. Each x-server is within one node.

An x-system may consist of

a) a single x-server;
b) several non-interacting x-servers;
c) several interacting x-servers.

The interaction between an x-client and an x-server is governed by an x-access-protocol (see figure D.6). These protocols are subject to specific x-application standardization and are not addressed in detail in this part of ISO/IEC 10031.

Several x-servers connected through a network may interact to form the total x-system. In that case, they co-operate by means of an x-system-protocol. These protocols are subject to specific x-application standardization and are not addressed in detail herein.

The subset of the x-access-protocol available between the x-client and a particular x-server depends on the x-service-definition and the partitioning of the x-server. For x-servers which are maintaining a distributed information base, for example, the x-access-protocol may contain a referral mechanism (which means that the contacted x-server returns a hint to the x-client as to which other x-server(s) to contact in order to find a particular piece of information), if an x-system-protocol does not exist, or if the x-server is not not able, or unwilling to obtain this in a way transparent to the x-client.

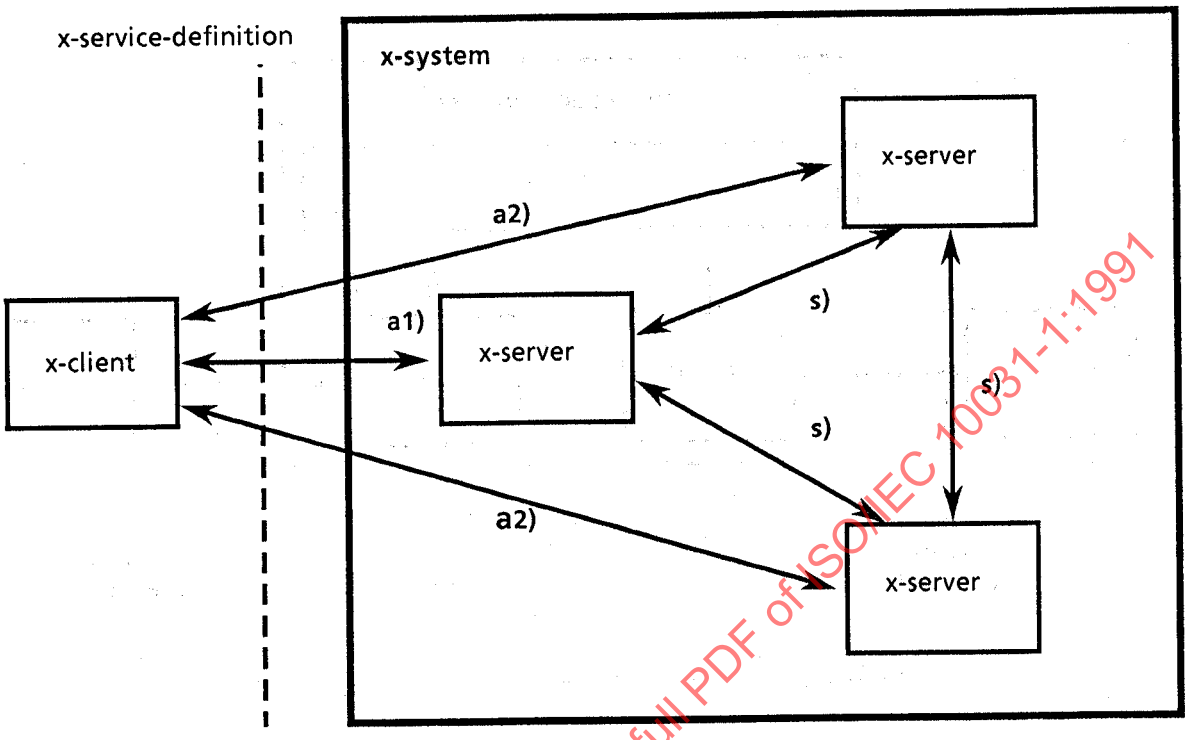


Figure D.6 – X-system

Note that an x-client is not introduced for x-system-protocols, although the same OSI concepts of application-process and application-entity, etc., are applied. The inherent asymmetry of the client/server model may not be useful in the design of some system protocols.

D.3.4 Cooperation between servers

D.3.4.1 A Server as a user of another server

Sometimes one system (x-system) will use another system (y-system). This is modelled by describing the x-server that needs to use the y-system as having the role of a y-user of the y-application. In this case, a y-client exists within the application-process containing the x-server.

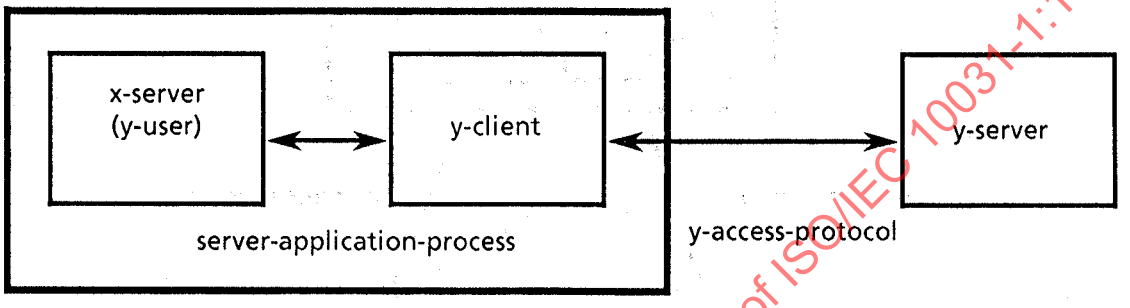


Figure D.7 – A Server as a User of Another Server

This model can also be used when the two servers are of the same type, i.e., when an x-server is using another x-server.

D.3.4.2 Referenced-object-access

For some data object types (see D.2.4) an x-server acts as a source and a y-server acts as a sink of data values (e.g. a Message Store server may be a source and a Printing server may be a sink of printable documents such as Interpersonal Message Body Parts). In general, a server may act as both a source and a sink of data (e.g. Document Filing and Retrieval Server, Message Store Server).

If an x-client and a y-client are co-located within one application-process and an object value has to be transferred from an x-server to a y-server, it may be inefficient to transfer this object value via the x-access-protocol from the x-server to the x-client, and then via the y-access-protocol from the y-client to the y-server (see figure D.8). In this case it is more efficient to transfer only a reference to the object value in the access protocols. The referenced object value itself is transferred directly from the source x-server to the sink y-server.

When an object value is transferred with an x-access protocol between the x-client and the x-server, it may be useful for the user-application process to manage the data transfer using a Distinguished-object-reference (DOR).

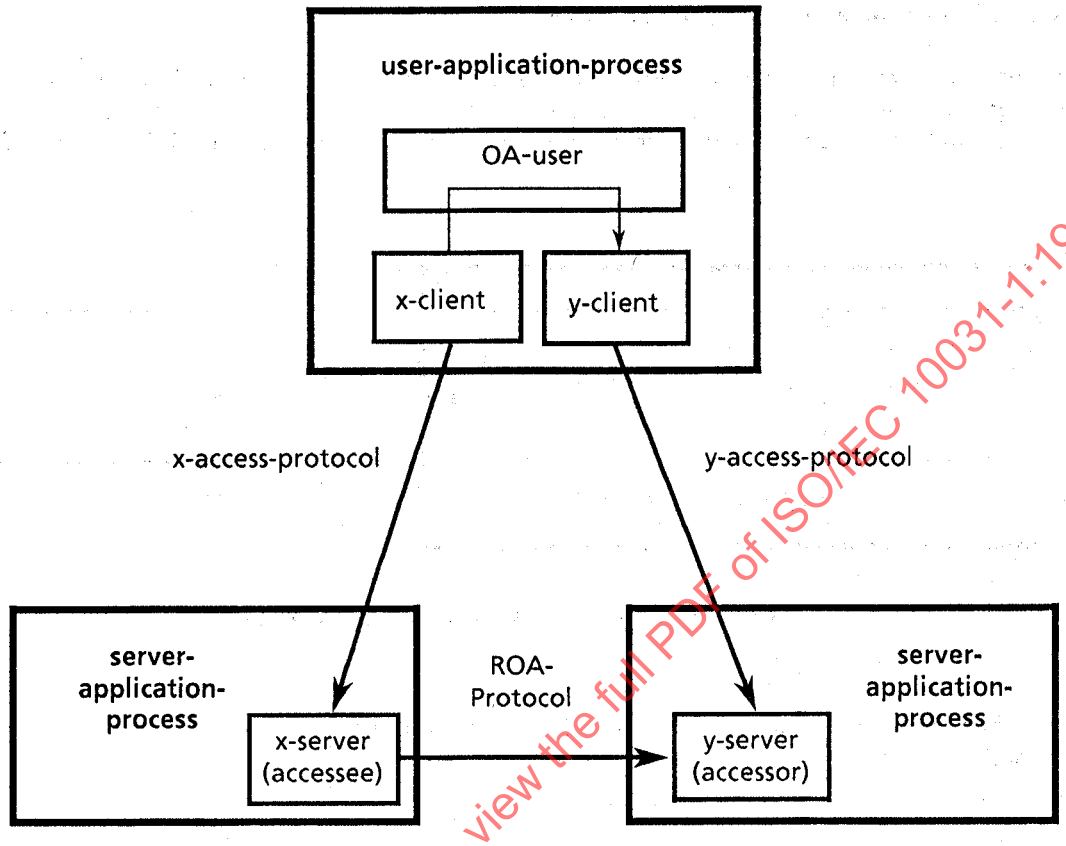


Figure D.8 – Referenced-object-access

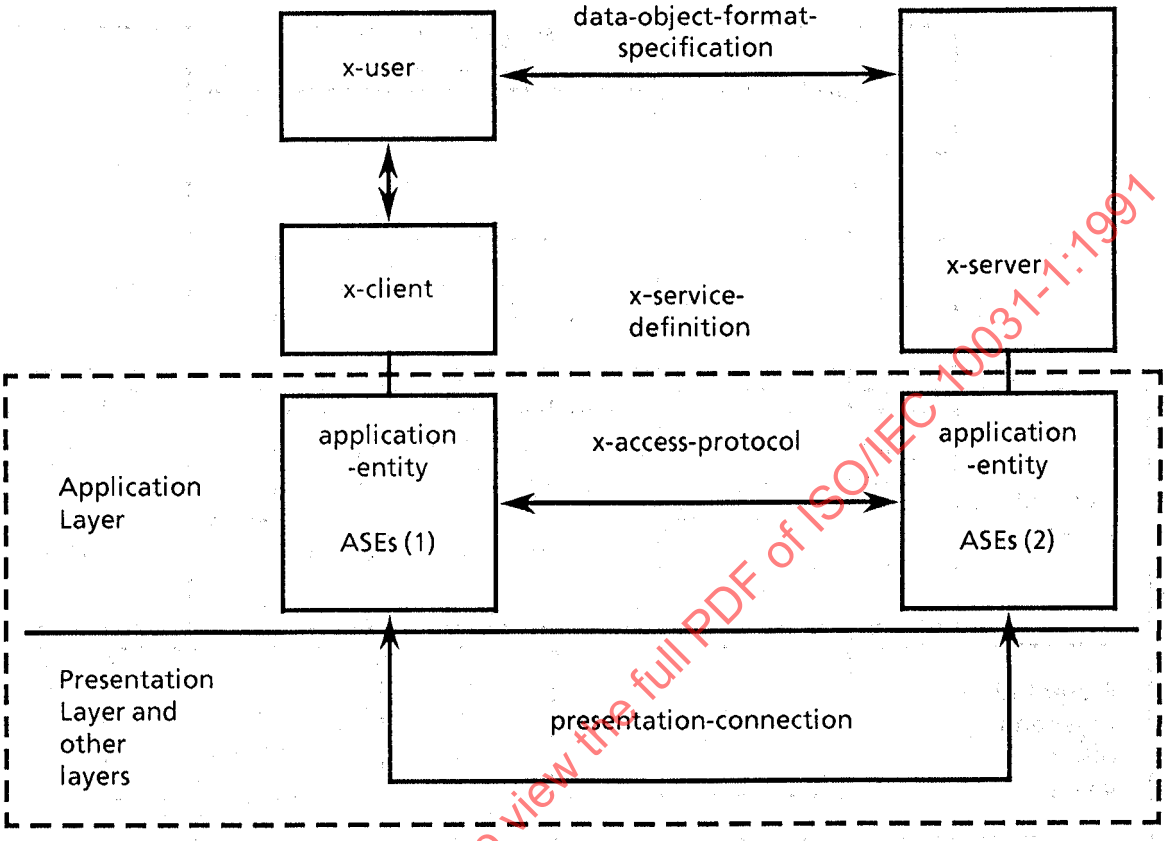
D.4 Client-server communication model

This subclause specifies some additional details about the communication between an x-client and an x-server, on the basis of the model introduced in D.2.3 and figure D.3.

Figures D.9 to D.12 show various examples of configurations that require different sets of application-service-elements. The Association Control Service Element (ACSE) is required in every set of application service elements. Furthermore, the Remote Operations Service Element (ROSE) is required in every set. The Reliable Transfer Service Element (RTSE) is optional. Which additional application service elements are required in a given set depends on:

a) the nature of the distributed-office-application concerned;
b) whether the set is associated with a client or a server;
c) whether the set implements the access protocol or the system protocol.

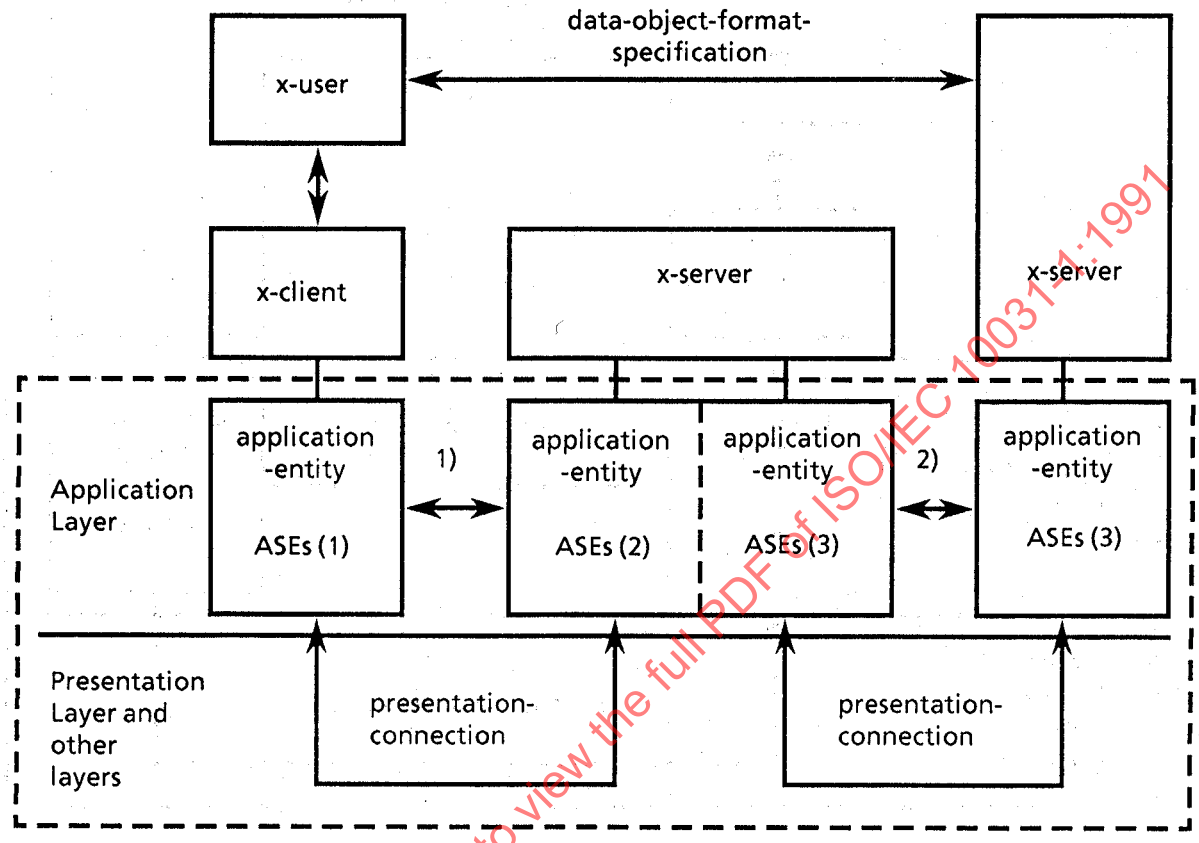
The data-object-format-specification represents a basis of cooperation between the x-user and the x-server, or between OA-users.



ASEs (1) This set of application-service-elements implements the functions required by an x-client for communication with an x-server, using the x-access-protocol.

ASEs (2) This set of application-service-elements implements the functions required by an x-server for communication with an x-client, using the x-access-protocol.

Figure D.9 – OSI Communication between an x-client and an x-server



ASEs (1) This set of application-service-elements implements the functions required by an x-client for communication with an x-server, using the x-access-protocol.

ASEs (2) This set of application-service-elements implements the functions required by an x-server for communication with an x-client, using the x-access-protocol.

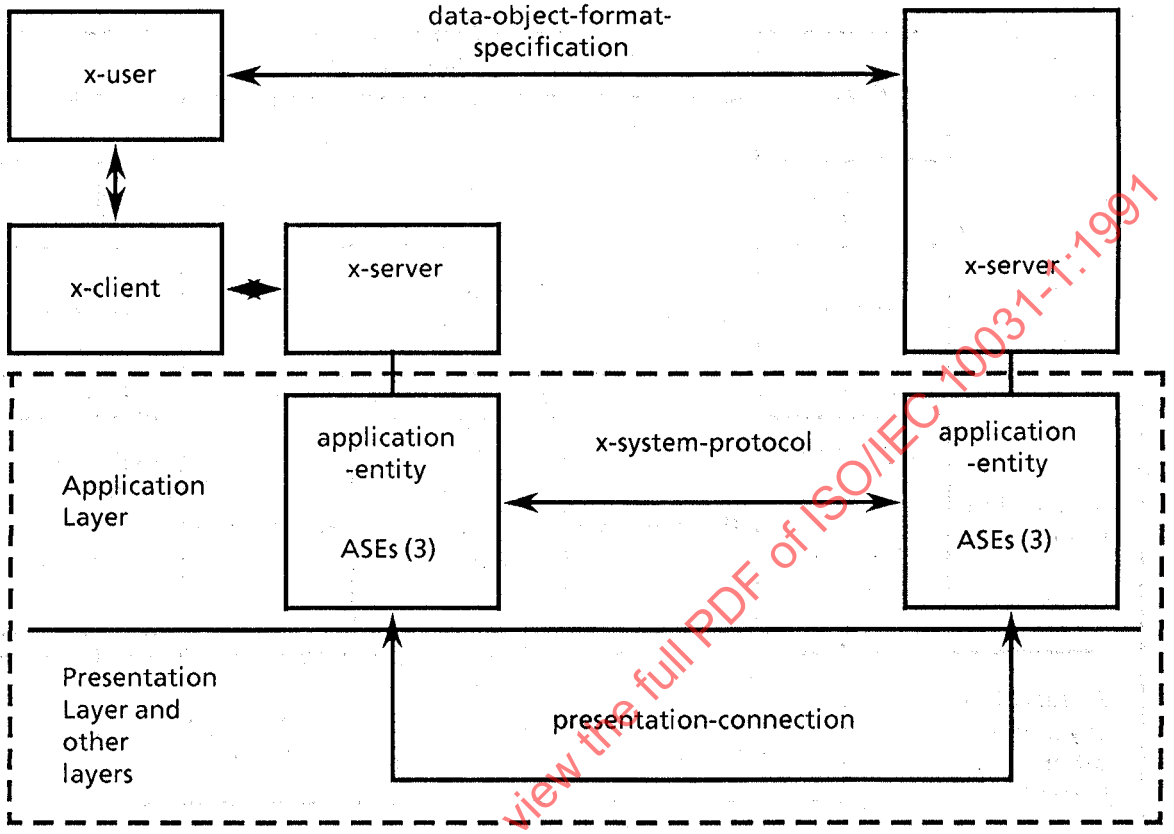
ASEs (3) This set of application-service-elements implements the functions required by an x-server for communication with another x-server using the x-system-protocol.

1) x-access-protocol.

2) x-system-protocol.

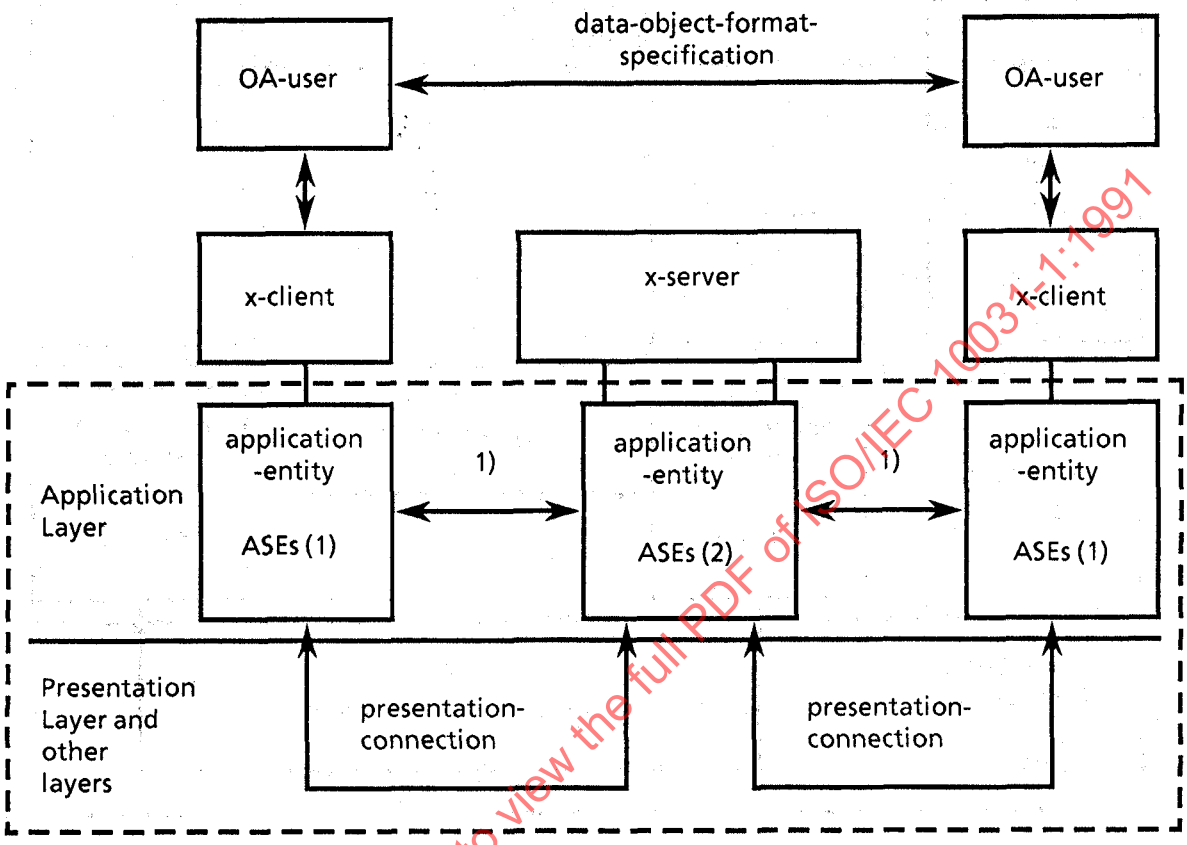
NOTE – The application-entities including ASEs (2) and ASEs (3) respectively may be combined into a single application-entity supporting two application contexts.

Figure D.10 – OSI Communication between an x-client an x-server and between two x-servers



ASEs (3) This set of application service elements implements the functions required by an x-server for communication with another x-server using the x-system-protocol.

Figure D.11 – OSI Communication between an x-client with a co-located x-server and another x-server



ASEs (1) This set of application-service-elements implements the functions required by an x-client for communication with an x-server, using the x-access-protocol.

ASEs (2) This set of application-service-elements implements the functions required by an x-server for communication with an x-client, using the x-access-protocol.

1) x-access-protocol.

NOTE – In figure D.12 the x-server is used as a store-and-forward system between the two OA-users (e.g. in Document Filing and Retrieval or Message Transfer). In this case the data-object-format-specification represents the basis for cooperation between two OA-users.

More than one access protocol may be defined in a single application context.

Figure D.12 – OSI Communication between two x-clients and an x-server and between two Users

D.5 Functional categories

D.5.1 Productive and supportive applications and facilities

A distinction is made between supportive applications and productive applications and between supportive roles and productive roles of an application.

It is worth noting that the categorization of applications as "productive" or "supportive" is somewhat imprecise, though nonetheless useful. For example, the basic Message Transfer application could potentially be used as a supportive application for other applications, such as for distributing directory updates between directory servers. Likewise, although the Directory application is generally viewed as a supportive application, it could also be considered as a productive application when used for responding to human user queries for information. Rather than being an intrinsic property of the application, the distinction is based on whether or not an application directly provides facilities of interest to human users.

Applications with supportive roles collaborate in order to provide users with a stable, "high-level" environment. Just as a programming environment consists of a number of programs (many of which are general utilities), the network operating environment for the productive distributed-applications consists of several supportive applications. These supportive applications, which are built using the same model concepts as the productive applications, constitute the general operating support that can be assumed by the productive applications, and provide the users of the OSI distributed-office-applications with a "high-level" view of their environment, such as allowing them to be decoupled from the location and physical addressing of various devices and resources.

In other words, these supportive applications constitute the high level support environment for the productive applications and for the users of these applications.

The productive applications are visible to the human user, are seen as useful, and are specifically used by him. For the general office worker the productive applications include remote printing, document filing and retrieval, mail, some uses of directory, etc.

D.5.2 Operating support

The general operating support that can be assumed by the productive applications includes, for example:

a) time base;
b) authentication and attribute facility;
c) some directory functions (e.g. name to address mapping).

This list is not exclusive. Annex H gives a more detailed view of these roles.

D.5.3 Management

Some management functions are particularly important, e.g. those functions recording operational behaviour of the distributed-office-applications environment.

Other management functions are seen as distinct applications, and are seen by the human user as productive applications. This is particularly so for the analysis and presentation of management information.

Management may be the subject of future standardization.

D.5.4 Guidelines for applications

Some supportive applications, for example, authentication, have a major impact on other access protocols. This is both in terms of the information carried by the protocols and in terms of the sequence in which operations are performed. The framework will specify in due course, for instance, the permitted sequences required for authenticating, requesting and gaining access to a server. The area of authentication may be the subject of future standardization.

Other functionality, for example logging and accounting, have an impact on application design and specification. The areas of security logging and accounting may be the subject of future standardization.

Administrators of a distributed office application system will choose policies for some of these aspects; so productive applications will need to be adaptable to changes in these policies.

Clause 6 of this part of ISO/IEC 10031 gives an initial set of guidelines, which can be used at present, until certain aspects e.g. the security aspects, have been studied more in depth.

D.6 Types of interactions between applications

This sub-clause describes and classifies different types of interactions as they relate to the categories of supportive applications and productive applications introduced in clause D.5.

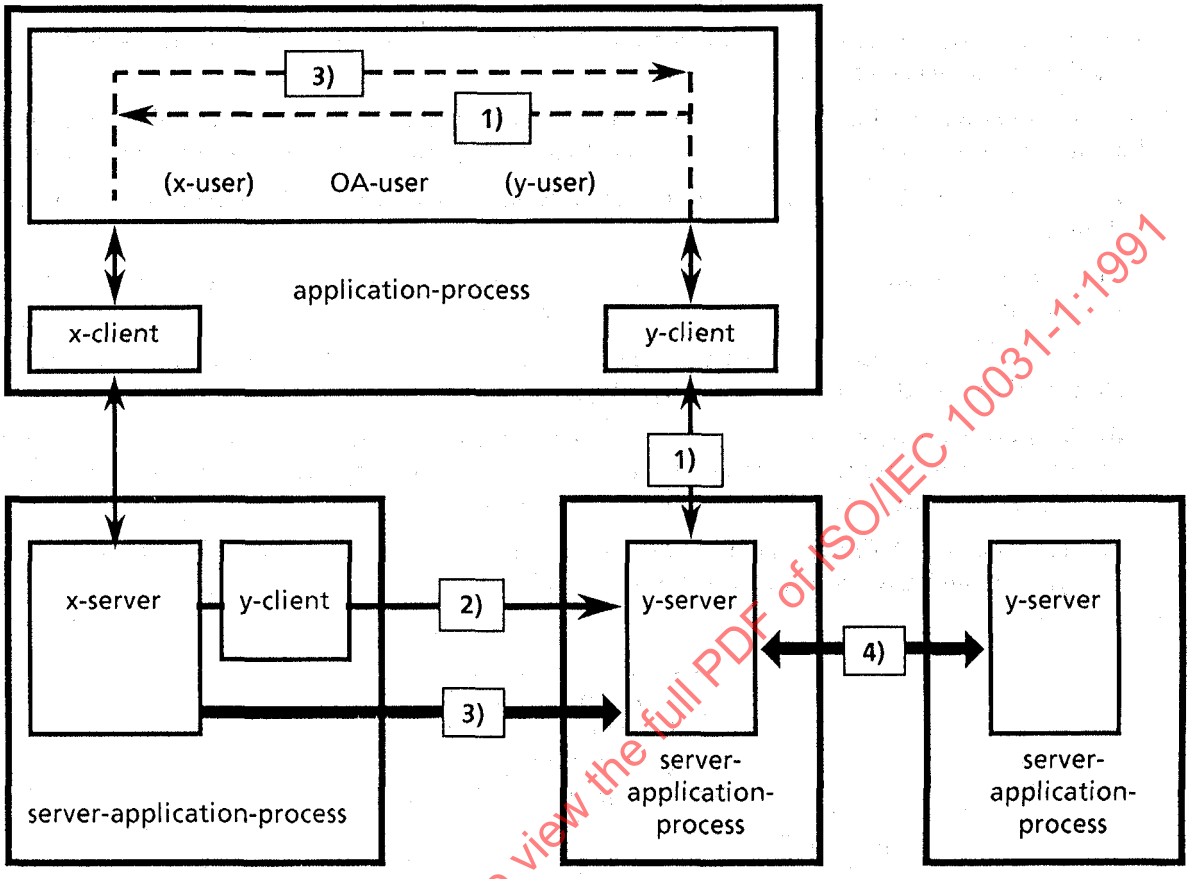
These types of interactions will be used in later subclauses of this annex.

The interactions between an x-user and a productive x-application is described in the former part of this annex and is not repeated here.

The interactions between, e.g. an x-user and a supportive application may occur in relation to an interaction between the OA-user and a productive application. This is shown in figure D.13 as a Type 1 interaction between the OA-user and the (supportive) y-application. Information obtained in this Type 1 interaction is used by the OA-user in the interaction with the (productive) x-application. The interactions uses the x- and the y-access-protocols respectively.

The interactions between two servers (either of which may be productive or supportive) is shown as a Type 2 interaction in figure D.13. The x-server uses the co-located y-client to access the y-server using the y-access-protocol.

Finally a set of coordinated interactions exists that is known as a referenced-object-access. Here the user or an entity acting as x-user and y-user using the x- and y-access-protocols, instructs the x-server and y-server to perform an information transfer.This is shown as a Type 3 interaction in figure D.13. The Type 3 interactions thus contain two coordinated actions. First, the action, internal to the user, to coordinate the setup for a referenced-object-access with the x-server and the y-server, and secondly, the required access itself.



1) Type 1 interactions between an OA-user or a server acting in an x-user role and a y-server which result in information obtained from the y-server being used in interaction with the x-application.

2) Type 2 interactions between an x-server and a y-server using the y-client and the y-access-protocol.

3) Type 3 interactions between two servers acting upon instructions issued by the OA-user or an entity acting in the role of x-user and y-user (using the x-access-protocol and the y-access-protocol). The information transfer from the x-server to the y-server uses a ROA-protocol.

4) Type 4 interactions between two servers of the same type, using a system-protocol defined for that purpose. Figure D.13 shows two y-servers using a y-system-protocol in a Type 4 interaction.

Figure D.13 – Types of Interaction

D.7 Example of application interactions

Figure D.14 shows an example of interactions between an OA-user and applications performing productive and supportive functions.

The simple action of filing a message received in the Message Store will actually require the following operations once the OA-user starts with a bind-operation to the Message Store server.

a) The Message Store server accesses the Authentication & Security Attributes server (type 2 interaction).

b) The OA-user accesses the Message Store server to identify the message as an object to be offered later.

c) The OA-user accesses the Directory to get the address of the Document Filing and Retrieval server able to file the message (type 1 interaction).

d) The OA-user accesses the Document Filing and Retrieval server to select the structure under which the message has to be filed and identifies it with the reference supplied as a result of the previous interaction 2).

e) The Document Filing and Retrieval server gets the message from the Message Store server through ROA-protocol (type 3 interaction).

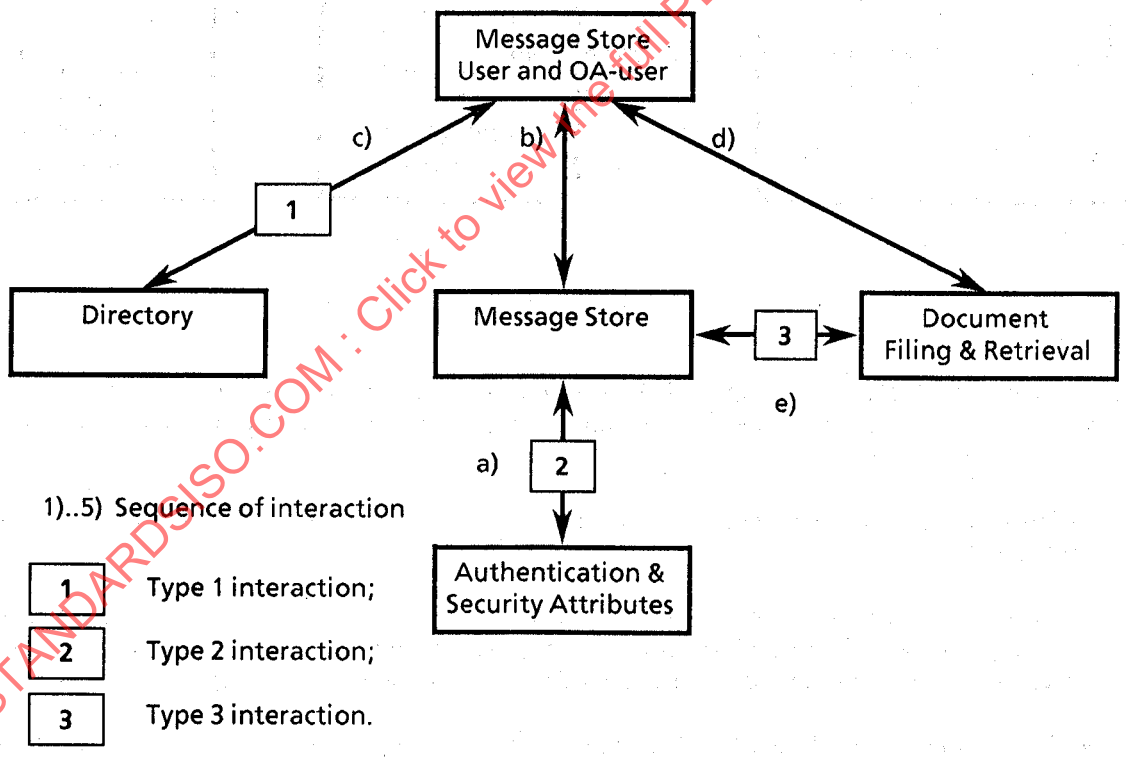


Figure D.14 – Interactions between an OA-user of the Message Store and other applications

Annex E
(informative)
Identification considerations

E.1 General requirements

The distributed-office-applications environment is characterized by the geographical and logical dispersion of entities such as applications, nodes, objects, clients and servers. These entities must all be made to work together in a coherent interconnection in order for a distributed-office-application to effectively carry out its tasks.

An important tool in developing this coherent interconnection is the utilization of the concept called "Naming". A "Name" is a linguistic concept that identifies a particular entity from among the set of all entities. An entity, e.g. a server, would therefore have its own name distinguishing it from other servers. This distinguished name would be found within the Directory application.

Some system entities which will have distinguished names are listed below.

- Human user;
- Group of human users;
- Node (user node and server node);
- Server;
- Data Object (e.g. Document Store).

In addition to identifying the separate entities, the naming concept assists in providing other functional capabilities necessary for distributed-office-applications. User nodes and server nodes need, for example, to be able to find the names of:

- a Filing and Retrieval server that contains a specific filing cabinet;
- a server that contains a Message Store for a specific user;
- a server that holds the master copy of data (e.g., that one who is allowed to update a part of the directory);
- Print servers in an organization that can offer Elite typefaces and wide carriages.

Users can accomplish these actions because the Directory offers the following functional capabilities:

- Name-to-attributes binding: This capability binds a name to a piece of information related to the entity (object) to which the name refers. Name-to-address binding is a special case of name-to-attribute binding. The large number of entities that may reside in a distributed-office-application system, make this capability essential. This function is analogous to a "white pages" directory.

- Attribute-to-set-of-names binding: This capability lists the name or names of entities (objects) which have a given attribute or attributes. One example of this is a "yellow pages" directory which can be used, for example, to find the names of all remote Print servers in an organization which have "Elite" typefaces and have wide carriages available to them.

- Alternative names: Different names for the same entity (object), i.e. aliases, are useful in allowing users more flexibility to access the numerous entities of the distributed-office-application environment. In order to provide different spellings of the same object, it may be convenient to introduce an alias (e.g. MUNICH / MÜNCHEN). Similarly, in an inter-enterprise environment, a server may be referenced externally by a name different to its internally used one, to hide any organizational details.

E.2 Name Types

E.2.1 Overview

Standards for distributed-office-applications require the following naming and addressing information:

a) names and addresses required for control of application associations;
b) names required for identification of human users and x-user;
c) names required for data objects.

E.2.2 Names and addresses for control of application associations

ISO 7498-3 sets forth some principles for naming and addressing. Names and addresses required for the control of application associations are defined in ISO 8649 (ACSE).

E.2.3 Names for users

In the context of MOTIS, users are identified by O/R names. An O/R name comprises optionally a distinguished name.

In the context of the other distributed-office-applications users are identified by distinguished names.

O/R names are defined in ISO/IEC 10021-2, distinguished names are defined in ISO/IEC 9594-2.

E.2.4 Names of objects

If information about objects is stored in the Directory (e.g. a document store) these objects are identified by distinguished names (see ISO/IEC 9594-2).

There can be rules for identifying objects within a collection of objects, for example to identify a document within a document store.

Other objects requiring unique identification (see annex, clause E, 3) are identified by ASN.1 object identifiers.

E.3 Registration of identifiers

A number of object types will have to be given standardized identifiers, either as part of the respective standards or by some registration authority. A few examples are given below. Identifiers are OBJECT IDENTIFIERs as defined in ISO 8824.

E.3.1 Application context

The Association Control Service Element (ISO 8649) requires identification of Application Contexts.

E.3.2 Message content types

The message content types defined in MOTIS identify the different types of content that can be carried by the P1 protocol. Content type is an example of the data-object-format-specification (figure D.9). For details see ISO/IEC 10021-2.

E.3.3 Message body part types

Message body part types identify the different types of format/encoding which can be found as part of a User Message (see ISO/IEC 10021). These message body part types are also data-object-format-specifications.

Some message body part types are defined in ISO/IEC 10021-7. Others may be defined in other International Standards or by registration authorities.

E.3.4 DOR object types

DOR object types need to be identified (see ISO/IEC 10031-2).

E.3.5 Attribute types

Attribute types will be defined by individual distributed-office-applications standards, using the concept of "Attributes" (see 6.4.5). If an attribute type identifier has been allocated for a particular type of information in one application, this type can (and should) be used by other applications needing the same attribute type.

Annex F
(informative)
Security concepts

F.1 Introduction

This annex is tutorial in nature.

F.1.1 A definition of "Security"

For the purposes of this part of ISO/IEC 10031 "security" refers to characteristics of office systems that give resistance to accidents, failure and misuse, intentional or otherwise. Thus, security refers to a complex of procedural, logical and physical measures aimed at prevention, detection and correction of certain kinds of accident, failure and misuse together with tools to administer and manage these measures.

Given this definition, security does not only address intentional misuses, e.g. threats to a system, but it also addresses accidents such as the misrouting of a message and pinpointing the cause of the misrouting so that the responsible party can be identified.

In this way, security improves the integrity of doing business in addition to addressing threats to the organization itself.

F.1.2 Scope of security

Many different security needs imply a common set of secure functions to be provided independently of office applications. These common, secure functions will become visible in the interactions between users and productive applications, between productive applications and supportive applications but also in the installation, maintenance and management of applications and of the underlying system. These functions, their interactions and their management constitute the scope of security in this part of ISO/IEC 10031.

F.1.3 Security policies

To be effective, security measures need to be coherent. Therefore, an organization will define its security measures and methods of administration and management of those measures in a security policy. The responsibility for executing the security policy and for maintaining its effectiveness is exercised by a security-administrator.

Below are examples of security measures to be addressed by a security policy. Which measures to implement as part of a given security policy depends on the environment of the organization:

a) integrity of information contained in and/or processed by a system
b) confidentiality of (selected) information contained in and/or processed by a system;
c) integrity of services and functions provided by a system;
d) confidentiality of services and functions provided by a system.
e) means to obtain third party guarantees for certain operations. In other words, verification of the integrity of processes and information by third parties is needed;
f) means to authenticate individual users or groups of users according to defined rules;

g) control of access to servers, functions and information available on or through a system;
h) control of the flow of information within and between systems.

In the general case the organization will require interaction with other organizations. Organizations will select their own policies; each security policy can be said to apply to a given security-domain which is under the control of a single security-administrator. Doing business requires, by implication, interaction between security-domains. This too needs to be addressed by a security policy.

In some cases, two security-domains may interact directly, in other cases they may interact trough a third party. Also, the degree of trust between security-domains may vary.

F.2 Security Requirements for Distributed-office-applications

F.2.1 General security requirements

This subclause introduces general security requirements as these occur in a distributed-office-application environment. These requirements reflect both implied requirements - for example no system can be secure without some form of control of access - as well as specific requirements for security functions from a user point of view - for example, authentication of data origin.

F.2.1.1 Protection of access

F.2.1.1.1 General

Access control provides the means to confine access to certain known users as well as to control access by these users to specific resources for specific operations. Thus, control of access has two major components: authentication of users and authorization of access by authenticated users. Access control will be exercised according to an access-control-policy which applies to a security-domain.

F.2.1.1.2 Authentication

Users gaining access to a distributed-office-application system will be authenticated before being allowed access to any particular application that is subject to access-control-policy. Users may also require that the servers accessed are authentic. Users may access an x-server from a node belonging to the same security-domain as the x-server or from a node belonging to another security-domain. In either case, a commonly agreed procedure of exchanging authentication information must be used.

Authentication may be time bound; repeated authentication throughout an occasion of communication may be required by certain security policies.

F.2.1.1.3 Access authorization

Nodes in a distributed-office-application environment may require the use of access authorization to protect the confidentiality and integrity of security-objects and the integrity of the server node. Authorization methods may use a variety of mechanisms such as access control lists, capabilities and other security-attributes, singly or in combination.

Users will be granted access to x-servers and to security-objects within x-servers based on their privilege-attributes under the prevailing security policy of the security-domain(s) involved.

Whenever users access servers or security-objects not belonging to their security-domain, authorization information as required by the serving security-domain, will be passed in a secure fashion.

F.2.1.2 Protection of data information

Security policies may require data interchanged with or stored on a distributed office system to be protected from external attack. "External" in this context is used to indicate other than by the normal system access route, (e.g. line tapping, theft of media.)

Data protection covers both confidentiality (keeping secret) and integrity (protecting against change).

Within a distributed-office-application environment the following requirements with regard to data protection apply:

a) Protection of data in storage (even on removable media);
b) Protection in interchange, e.g. access control information, messages, electronic documents and files exchanged between systems.

Protection refers to preventing leakage of sensitive information as well as preventing contamination of trusted information with untrusted information.

Unless physical protection is to be depended upon, confidentiality may require the use of encryption, integrity may require the use of digital signature.

Encryption techniques require the use of cryptographic keys. Systems supporting encryption must provide a secure method of key management both within a security-domain and between security-domains.

F.2.1.3 Protection of usage of resources

Security policies may require protection of usage of resources. This protection takes two forms: keeping usage secret (confidentiality of usage) and preventing denial of service.

F.2.1.4 Accountability of usage of resources

Security policies may require means to assure accountability of usage of resources. Accountability includes the selective logging of an audit trail of operations (both attempted and completed) as well as non-repudiation of data origin and of receipt.

Non-repudiation is proof, a posteriori, to a third party, of the identity of an entity that sent or received for example a given message. It is closely allied to data integrity and is usually combined with it.

F.2.2 Security management requirements

F.2.2.1 General considerations

Systems supporting secure distributed-office-applications should provide the operating organization with the tools to manage the security-facilities of these systems. Examples of these tools are secure software installation and audit facilities for auditing the operation of the security-facilities.

Users of a distributed-office-application trust the integrity of system components to perform the expected functions and no other.

F.2.2.2 Aspects of security management

For every kind of security function defined for the distributed-office-application environment, four aspects of misuse or breach of security should be addressed that together define the management requirements of these functions. These aspects are: prevention, detection, recovery and administration. Depending on the level of security desired, some or all of these aspects become visible in actual implementations.

PREVENTION is based upon rules for managing a security function or application. An example of such a rule is changing passwords every three months.

DETECTION is based upon auditing of the security operations of a system.

Auditing of a security related operation provides security-administrators with feedback concerning the use and effectiveness of the security functions of the system.

Auditing has three components:

a) audit trail generation and collection;
b) audit trail analysis; and
c) audit trail archiving.

The latter belongs to the aspect of administration.

In a distributed-office-applications environment an application may be distributed over multiple security-domains. Where such is the case, commonly agreed audit techniques may be needed to facilitate inter-domain cooperation.

RECOVERY of a security breach - real or suspected - may require changes in security procedures and information available at the different nodes of a distributed system. Therefore, protocols and procedures are needed to support the implementation of recovery measures.

ADMINISTRATION has two aspects related to the life of the system:

a) gathering information from the system;

b) creating information for the system.

The first aspect concerns reports made from information logged in the protected data base. Special filters must be provided so that a security-administrator can adapt the report to get only the kind of information he needs. The second aspect deals with the creation or the deletion of security-subjects and security-objects and with the definition of keys, rights and passwords (at least the initial password).

F.3 Secure systems model

F.3.1 Overview

In a secure distributed system, a number of activities must be engaged to provide that security.

The secure systems model divides these activities into elements, each one having a single, coherent role to play in the provision of the total security picture. These abstract elements are intended as reasoning tools rather than as real implementations of security functions. These elements are referred to as security-facilities.

Having identified the security-facilities and the communications between them, it can be shown how they might combine together to form supportive security applications or how they become trusted components of user-application-processes and server-application-processes, and standard protocols defined where appropriate for their interactions with each other and with their elements of the distributed-office-application environment.

In terms of the OSI model, the level of view addressed here is above the Application Layer. The supportive security applications described communicate using services of sufficient security to satisfy their needs. These needs take the form of guarantees, to some acceptable level, that communications between them and with their untrusted peers are confidential and unmodified, and that each communication is with a known and identified peer entity.

The model for the provision of these guarantees at lower OSI layers may be a subject for future standardization.

There are two fundamentally different levels at which the security requirements of a distributed system need to be addressed:

a) application independent level, to control access to distributed system security-objects as user-application-processes, server-application-processes, workstations, communication resources, etc.;
b) application specific level, to control access to specific security-objects within an application (such as a document).

These two levels of view have quite different requirements reflected in different security policy subsets tailored to the different kinds of protected security-objects involved and the different components that are responsible for their support. Something that is considered as a protected security-object at one level can become an accessing security-subject at another.

F.3.2 Security-facilities

At this stage of description, the reader should make no assumptions about the degree of distribution of the facilities; this might vary from being a single security server to being an aspect of every distributed supportive or productive application. Neither is it suggested that all of these facilities need be available on every node of a distributed system. They should be viewed as a list of building blocks from which a choice can be made appropriate to the security policy and level of security required for the distributed system. However, by identifying the full list, the model causes omissions to be made evident and any resulting security weaknesses other than accidental. Annex F, 3.3 and annex H clause 2 show some of these security-facilities combined into three supportive security applications. This part of this annex identifies the following security-facilities:

F.3.2.1 User sponsor facility

The only entity in a distributed system that is aware (independently of any services that may be being used) of an individual security-subject's current access to protected security-objects. The security-subject involved will usually be a human user, but under policies where servers access to other servers is policed, the security-subject can be an x-server. Its responsibility include:

a) passing credentials for authentication;
b) initiating of service selection;
c) timing out inactive users.

F.3.2.2 Authentication facility

Accepts and checks security-subject credentials, communicating its conclusions to other security-facilities. The security-subject will either be a human user via his user sponsor, a non-security application acting as security-subject (i.e. an x-server using a y-server), or a non-security application coming on-line and making itself available.

F.3.2.3 Security-attribute facility

Provides appropriate subject-related access privilege-attributes and access control-attributes, to be used to authorize or deny requested access by security-subjects to security-objects.

F.3.2.4 Authorization facility

Uses access-context, (security-subject) privilege-attributes and (security-object) control-attributes to authorize or deny requested access by security-subjects to security-objects.

F.3.2.5 Association management facility

This facility ensures:

a) secure underlying communications, including assurance of the identity of the communicating entities.
b) authorization, via an authorization facility, for the two entities to communicate on behalf of the controlling user.

How the upper layer architecture functions relate to or may be used to support the Association Management Facility may be the subject of future standardization.

F.3.2.6 Security state facility

Maintains the current dynamic state of authenticated security-subjects and security-objects in the distributed system, their associations and the privilege-attributes carried by those associations.

F.3.2.7 Security audit Facility

Receives event information from other security-facilities for recording and immediate or later analysis.

F.3.2.8 Security recovery facility

Acts upon event information from the Security Audit Facility according to a set of rules defined by a security-administrator.

F.3.2.9 Inter-domain facility

Controls and maps one security-domain's interpretation of security-subject identity, security-object identity, authentication and authorization data into another security-domain's

interpretation. Helps Association Management form associations between entities in different security-domains.

F.3.2.10 Cryptographic support facility

Provides cryptographic functions used both by other security-facilities and applications to secure data in storage and transit in the following specific ways:

a) confidentiality of data;

b) confidentiality of communications;

c) integrity of communications;

d) data origin authentication;

e) non-repudiation of origin;

f) non-repudiation of receipt.

F.3.3 Supportive security applications

For the purpose of this part of ISO/IEC 10031, the following three security applications are identified:

a) Authentication and Security-attributes application. This combines the Authentication Facility and the Security-attribute Facility;
b) Inter-domain application. This is the Inter-domain Facility;
c) Security Audit application. This is the Security Audit Facility.

In addition other supportive security applications may be defined that implement other combinations of the security-facilities given in annex F, 3.2. Note that the presence of security-facilities, e.g. Association Management, whether as separate entities or as part of user-application-processes and server-application-processes will need additional protocol elements for distributed-office-applications.

F.3.4 Proxy

In some cases an x-server may be accessed by a y-server rather than directly by a user. There are two situations:

a) the initiating x-server is acting on its own behalf; or
b) the x-server is acting on behalf of of another security-subject (e.g. a human user).

The first situation may be used for example to restrict access to security-objects held on one server (say File server) to those coming via another (say Database server). It is entirely appropriate for the Database server to act with respect to the File server as a security-subject with its own identity and access privileges.

On the other hand, it might be appropriate for the initiating x-server to act on behalf of the user (by proxy) and assume some or all of his security-attributes. The proxy could either imply trust by the user for a single specific access request or it can imply wider powers. The proxy may either contain access request details or it may contain a reference to those details. This may be the subject for future standardization.

In this way access can be controlled in terms of the route used.

F.4 Access-rights for distributed-office-applications

There are some DOA-specific security features. An example is the access-rights. Access-rights for Distributed-office-applications will be designed on the basis described in this sub-clause.

Table F.1 shows an example of the relationship between the standard set of abstract operations and one possible set of access-rights option.

Table F.1 – Access-rights and allowed operations

	OWN	RMD	RM	RO
List	x	x	x	x
Read	x	x	x	x
Modify	x	x	x	
Copy	x	x	x	x
Move	x	x		
Search	x	x	x	x
Create	x			
Delete	x	x		
Reserve	x	x	x	
Notify				
Abandon	x	x	x	x

X means that the corresponding operation is allowed under the corresponding Access-right

NOTE – Table F.1 shows the following four levels of Access-rights.

a) Owner (OWN);
b) Read-modify-delete (RMD);
c) Read-modify (RM);
d) Read-only (RO).

Annex G
(informative)
Management

Distributed-office-applications require procedural, logical and physical measures which provide the managers of distributed-office-applications with the ability to plan, organize, supervise, control and account for the use of the distributed-office-applications. These measures may operate on a single distributed-office-application or operate on multiple distributed-office-applications across a number of open systems. These measures are referred to as "management".

Management is provided by a range of facilities, each of which supports an aspect of the required measures. These facilities include:

a) fault management;

b) accounting management;

c) configuration and name management;

d) performance management;

e) security management.

General OSI Management is discussed further in ISO 7498-4.

Annex H
(informative)
Categories and relationship of applications

H.1 Introduction

This annex examines the general need for office applications, any one of which may have at various times, roles of a supportive or productive nature. In the former case, the application is supporting (provides a service) to another application. This other application will generally have a role of a productive nature, normally offering a service visible to a human user. This annex describes how cooperation is achieved between an application performing a supportive role and an application affording productive services.

NOTE – Hereafter in this annex the terms supportive and productive will be used to describe the role that any application is performing at the instance of time relevant to the activity being described. This annex will not presume that applications are inherently supportive or productive.

H.2 Operation of supportive applications and facilities

H.2.1 Time base facility

With the current proliferation of international organizations, a worldwide means of providing a reliable and unambiguous time base is required. This will become even more important in the future, as large numbers of nodes are connected to worldwide networks which themselves are interconnected to other networks.

The various components of any distributed system must be able to obtain the current time. This time could be used by other applications, e.g. to timestamp files, to timestamp messages, to enable authentication to be carried out.

Synchronization does not have to be exact but should maintain time to be within a reasonable spread (e.g. something of the order of 10 minutes) over the whole of the distributed system. The accuracy is set by the administrators.

More precise timing if required, (e.g for timestamps) is by use of local time facilities within a node. Values of time derived locally may need to be qualified by location information indicating the source of the time value.

The methods by which various hosts obtain and maintain the correct time is outside the scope of this part of ISO/IEC 10031.

This facility provides for a generalized international time containing day, month and year in the Christian era (with optional seconds), which can be specified to a precision of one second or one minute.

H.2.2 Supportive security applications

H.2.2.1 Introduction

This subclause describes the supportive security applications which together provide support for