TECHNICAL REPORT

ISO/IEC TR 10730

First edition
1993-04-15

Information technology — Open Systems Interconnection — Tutorial on Naming and Addressing

Technologies de l'information — Interconnexion de systèmes ouverts — Tutorial sur la dénomination et l'adressage

Reference number
ISO/IEC TR 10730:1993(E)

Contents

ISO/IEC Copyright Office • Case postale 506 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland.

Foreword

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

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC1.

The main task of a technical committee is to prepare International Standards, but in exceptional circumstances, a technical committee may propose the publication of a Technical Report of one of the following types :

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts ;

- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard ;

- type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ('state of the art', for example).

Technical reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful .

ISO/IEC TR 10730, which is a Technical Report of type 3, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Sub-Commitee SC 21, *Information retrieval transfer and management for open systems interconnection* (OSI).

Introduction

This Technical Report has been developed in order to answer JTC1 member comments on ISO 7498-3 requesting to provide tutorial material to give readers easy understanding. It has been decided to publish this tutorial as a Technical Report of Type 3 rather than as an annex to ISO 7498-3.

In this Technical Report, the basic concepts of naming, including the relationship between (N)-entities, (N)-service-access-points and (N)-addresses are developed in clause 5. A discussion of addressing information in services and protocols is then presented in clause 6, followed by layer-specific examples for the Application and Network layers. Registration authorities and directory facilities are then described in clause 7. Clause 8 presents a series of examples covering relationships between layers and the effects of both initiating and recipient mechanisms in Open Systems altogether with examples of specific name forms published in OSI standards.

Information technology - Open Systems Interconnection - Tutorial on Naming and Addressing

1 Scope

This Technical Report introduces the main concepts and mechanisms which are defined in ISO 7498-3 to fulfil the needs for Naming and Addressing (N & A) objects in the Open Systems Interconnection Environment (OSIE). It also includes the rationale for some of the important decisions made in the Naming and Addressing architecture.

Although ISO 7498-3 does not define any specific forms of names and addresses, this Technical Report concludes with examples of specific name forms that have been defined in other published OSI standards thereby showing how the concepts and mechanisms defined in ISO 7498-3 have been applied in the naming of certain objects.

2 References

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

ISO 3166:1988, *Codes for the representation of names of countries.*

ISO 6523:1984, *Data interchange - Structures for the identification of organizations.*

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

ISO 7498-3:1989, *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 3 : Naming and Addressing.*

ISO 8348 / Add.2:1988, *Information processing systems - Data communications - Network service definition - Addendum 2 : Network Layer Addressing.*

ISO/IEC 8824:1990, *Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).*

ISO/IEC TR 9577:1990, *Information technology - Telecommunications and information exchange between systems - Protocol identification in the Network Layer.*

ISO/IEC 9594:1990, *Information technology - Open Systems Interconnection - The Directory.*

ISO/IEC 9834-1:-1993, *Information technology - Open Systems Interconnection - Procedures for the operation of OSI registration authorities - Part 1 : General procedures.*

ISO/IEC 9834-6:-1993, *Information technology - Open Systems Interconnection - Procedures for the operation of OSI registration authorities. Part 6 : Application-processes and application-entities.*

ISO/IEC 10021:1990, *Information processing systems - Text communication - Message Oriented Text Interchange Systems (mOTIS).*

3 Abbreviations

For the purpose of this Technical Report, the following abbreviations apply.

AE	Application Entity
AET	Application-Entity Title
AEQ	Application-Entity Qualifier
AFI	Authority and Format Identifier
AFNOR	Association Française de Normalisation
ANSI	American National Standards Institute
AP	Application Process
ASN	Abstract Syntax Notation
ATDF	Application Title Directory Facility
BSI	British Standards Institute
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CL	Connectionless (Mode)
CO	Connection (Mode) (previously referred to as Connection Oriented Mode)
DCC	Data Country Code
DIS	Draft International Standard
DSP	Domain Specific Part
ECMA	European Computer Manufacturers Association
EWOS	European Workshop on Open Systems
FTAM	File Transfer, Access and Management
IAF	Initiator Addressing Function
IATA	International Airlines Transport Association
ICD	International Code Designator
IDI	Initial Domain Identifier
IDP	Initial Domain Part
IPF	Initiator PAI Function
JTC1	Joint Technical Committee 1
MHS	Message Handling System
NADF	Network Address Directory Facility
NSAP	Network Service Access Point
OIT	Object Identifier Tree
OSI	Open Systems Interconnection
OSIE	Open Systems Interconnection Environnment
PAI	Protocol-Addressing-Information
PCI	Protocol-Control-Information
QOS	Quality of Service
RA	Registration Authority
RAF	Recipient Addressing Function
RDN	Relative Distinguished Name
SAP	Service-Access-Point
SNPA	Sub-network Point of Attachment
TR	Technical Report

4 ISO 7498-3 and its relationship with this Technical Report

ISO 7498-3 states principles which are to be followed in any standard involving the need for identification and / or location of relevant objects within the OSIE. For the purpose of locating objects, a specific form of name (an address) is used.

Naming and addressing rules are essential to the success of OSI. In particular, it is a basic requirement that a real open system, however complex its internal structure may be, shows a simple naming and addressing structure to the OSIE, so that it may be easily accessible by any other real open system. ISO 7498-3 thus develops concepts which allow for the design of open systems which may have a very complex internal structure, while this complexity is not visible within the OSIE and the addressing scheme appears to be a very simple one from the view of other open systems. Such concepts preserve the principle of implementation independance (which is one of the basic rules of OSI), that is no real open system is required to know anything about the implementation design of any other real open system, nor does any real open system impose such knowledge as a condition for communication using OSI standards.

This Technical Report is intended to explain how ISO 7498-3 achieves this task. It is not intended to replace the base standard. Where any conflict arise with statements made in this Technical Report and those in the referenced standards, the base standards are the definitive source. The examples shown in this Technical Report are for explanatory purposes only and are not prescriptive.

5 Basic concepts

5.1 General aspects of naming

Names are linguistic constructs expressed in some language - i.e. names are composed of a given set of symbols. A name is bound to one or more objects. Within the OSI context names identify particular communication objects in the OSIE.

Naming of the objects considered in the OSIE may have global or local significance. The objects for which naming has global significance include real open systems and the elements of the OSI-layers (e.g. (N)-entity, application-process) .The addresses of these objects also have global significance.

The objects for which naming has local significance (i.e. significance within a given scope in an open system) include selectors, application-process-invocations and (N)-entity-invocations.

5.1.1 Types and properties of names

5.1.1.1 Primitive, descriptive and generic names

A name is unambiguous within a given scope when it identifies one and only one object within that scope. The unambiguity of a name does not preclude the existence of synonymous names for an object - i.e. more than one name can unambiguously identify an object. The concept of unambiguity may be extended to the case of the name of a set of objects.

Examples

- The name of a person (complete name) is unambiguous (generally) within the context of the family cell, but may often become ambiguous if this context is broadened. Other means are then necessary to ensure unambiguity, such as identity card/passport number, or social security number.

- IATA (International Airlines Transport Association) flight numbers are, generally, an example of unambiguous names.

- Within the OSI context, network-addresses are, by definition, unambiguous, as their purpose is to identify a set of NSAPs at the end system and, as a result, to locate the end system itself, among all possible end systems attached to any subnetwork.

Names can be categorised into primitive names and descriptive names.

A primitive name is a name that identifies an object (which may be a set of objects) and which is assigned by a designated authority. The internal structure of the name is not required to be understood or to have significance to users of the name.
A descriptive name is a name that identifies a set of one or more objects by means of a set of assertions concerning the properties of the objects of the set. The characteristic that distinguishes a descriptive name from a primitive name is that the structure of a descriptive name has significance to users of the name.

A descriptive name may be incomplete, in that many objects satisfy all the assertions or it may be complete in that it serves to identify a single object (e.g. a descriptive name may identify several FTAM-application processes and thus be an incomplete descriptive name. A complete descriptive name would in that case identify just a single FTAM-application-process).

A generic name is a primitive name or an incomplete descriptive name that identifies a set comprising more than one object. Note that when the membership of the set is not known to the user of the name, that user has no means to know if the name of the set is generic or not (e.g. a called-(N)-address (see 6.2) when used by the requesting system is viewed as a primitive name (regardless of whether or not it is generic) while this same called-(N)-address when processed within the responding system may be viewed as a generic name). A generic name can also identify the members (or a subset of the members) of an object class defined by an object type.

NOTES

1 ISO 7498-3 defines a primitive name as a name which identifies an object while implicitly recognizing that this object may itself be a set of objects. This implicit recognition arises from the definition of a generic name as "a primitive name which identifies a set of objects" (ISO 7498-3, subclause 5.6). A generic name is a specific case of primitive name where the fact that the object is a set is known.

2 Generally the intent when using a generic name for a specific action is that exactly one member of the set is selected as the target for the action (see ISO 7498-3 subclause 5.6). In such a case the requestor of the action is usually not aware of how the selection is made. Another recognised possible usage of generic names is when accessing a Directory Facility. In that case using a generic name as an input to the Directory Facility will result in the return of a list of the members of the associated set (see ISO 7498-3, subclause 14.2.3).

Examples

- In general family names are primitive names in that they do not convey information about the properties of the members of the family.

- IATA flight numbers are partially descriptive, as they are constructed as follows :

xxyyyy where : .xx is a 2-letter code identifying the airline
.yyyy is a code (up to 4 digits) identifying the xx company flight number

The two-letter code (xx) is an example of generic primitive name : it identifies the set of flights operated by this company (note that in order to ensure IATA flight numbers unambiguity it is sometimes necessary to provide extra information such as "leg of route").

- Sweepstake or lottery numbers are examples of unambiguous primitive (non-descriptive) names

- A subset of members of an application-process-type (with application-processes possibly located in different end systems) could be named by the generic name "MHS-ORGX". The member application-processes of "MHS-ORGX" could, for example, be all the MHS application-processes within a single organisation ("ORGX"). Each application-process would also be assigned a (primitive) name, say "MHS-ORGX-1", "MHS-ORGX-2",..., "MHS-ORGX-n". Using the generic name "MHS-ORGX" as input to the Application Title Directory Facility will result in the list of the associated application-process-titles ("MHS-ORGX-1", "MHS-ORGX-2",..., "MHS-ORGX-n") (see 7.3)

5.1.1.2 Titles and identifiers

A title is a name assigned to an object in order to discriminate among different objects (or sets of objects). Examples are application-process-title, application-entity-title, etc... A title can also be a name assigned to an object type in order to discriminate among different object types. Examples are application-process-type-title, application-entity-type-title, etc.

An identifier is a name assigned to an object in order to discriminate among occurences of this object. Examples of the use of identifiers are (N)-association identifiers, (N)-connection-endpoint identifiers, application-entity-invocation identifiers, etc.

5.1.2 Naming authorities and naming domains

Unambiguity of names is achieved through the use of naming authorities. A naming authority is a Registration Authority for names (see 7.1.3). Either it directly allocates and registers names (e.g. Network Addresses) or it just registers names submitted to it after having checked that they are not yet registered.The names registered by the naming authority should be expressed in a prescribed language and according to specific rules, but the naming authority does not perform the binding of a name to the object - or to the set of objects - it names.

A naming-domain is the set of names, that are assignable to objects of a particular type, and is administered by a naming authority. Naming-domains may be hierachically decomposed into subsets: the naming-subdomains. The naming-domain at the top of the hierarchy is the global naming-domain, which has the control of each naming-subdomain. The global naming-domain therefore is the set of all possible names - within the OSIE - for objects of a specific type. Independent global naming-domains may therefore exist for objects of different types. Each naming-domain is administered by a naming authority.

An object may be included in two or more naming (or addressing) domains. An object may also be allocated more than one name from a single domain. Thus there may be several names (or addresses) that identify (or locate) the same object. These names (or addresses) are synonyms. In the first case, synonyms are inconvenient but unavoidable in practice. In the other case, synonyms are usually useful and used intentionally (e.g. aliases and abbreviations).

Examples

- Today "Capital of France" is a synonym of the name "Paris - France" : these two names both identify the same object.
- Country names may also have synonyms of such type (e.g. United States of America,U.S.A,the States...).
- The term "OSI" is often used instead of "Open Systems Interconnection". Both terms are synonyms.

5.2 Names within the OSIE

5.2.1 Open systems

A fundamental component of the OSIE is the real open system, which is a system that conforms to the requirements of OSI standards in its communication with other real systems.

A system-title is used to identify a real open system. The system-title is a layer independent primitive name, i.e. it is used to identify a real open system as a whole. A single real open system is named by one and only one system-title. The system-title may be used in conjunction with other qualifiers to identify specific OSI resources in a real open system, i.e. it may be used as a basis to build structured names for objects in the given real open system. For example an application-process-title may be based on the system-title (with the addition of relevant qualifiers).

5.2.2 (N)-subsystems and (N)-entities

An open system is constructed of a set of layers. Each layer in a given open system defines one subsystem - the (N)-subsystem (for layer N). Therefore, an (N)-subsystem is an element in the hierarchical division of an open system (i.e. in the (N)-layer). An (N)-subsystem interacts directly only with elements in the (N+1) - and the (N-1) - subsystems of that open system.

An (N)-entity is an active element within an (N)-subsystem embodying a set of capabilities defined for the (N)-layer that corresponds to a specific (N)-entity-type. The (N)-entity-type can be identified but does not need to be located nor can it be located. Opposed to this, each of the (N)-entities (of that (N)-entity-type) can be identified and located ; this is necessary since the (N)-entities are the active elements which participate in the communication. An (N)-entity is named by an (N)-entity-title.

Since an (N)-entity represents communication capabilities of the (N)-layer, different communication capabilities of the (N)-layer may be represented by different (N)-entities - i.e. there may be several (N)-entities within an (N)-subsystem (e.g. two different (N)-protocols represented by two different (N)-entities).

When an (N)-entity is called to participate in the communication, there is a specific use of its functions - or parts of its functions. Such a use is called an (N)-entity-invocation. An (N)-entity-invocation is unambiguously named by an (N)-entity-invocation-identifier that must be unique within the scope of that (N)-entity.

The following names are used in conjunction with the (N)-entities :

- an (N)-entity-type is named by an (N)-entity-type-title ;
- an (N)-entity is named by an (N)-entity-title and ;
- an (N)-entity-invocation is named by an (N)-entity-invocation-identifier.

5.2.3 (N)-service-access-points ; (N)-SAPs

An (N)-entity is attached to one or more (N)-SAPs to provide the (N)-service to the (N+1)-layer. In order to do so the (N)-entity may use the service from the layer (N-1) provided through one or more (N-1)-SAPs.

An (N-1)-SAP is attached to one - and only one - (N)-entity, thus an (N)-entity is located by its attachment to one or more (N-1)-SAPs. Although an (N-1)-SAP-address strictly identifies an (N-1)-SAP, at any given point in time this (N-1)-SAP unambiguously addresses an (N)-entity.

Figure 1a and 1b illustrate the relationships of an (N)-entity to (N)-SAPs and to (N+1)-entities. It should be noted that the relationship shown in figure 1b is not allowed since an (N)-SAP can only be attached to one (N+1)-entity and one (N)-entity.

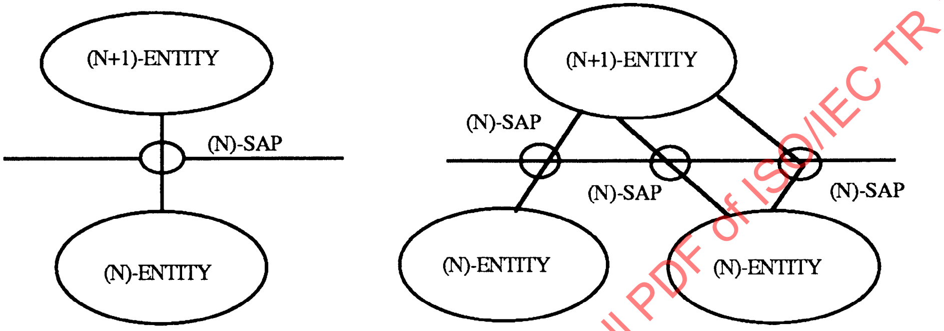


Figure 1a - Allowed relationships of (N)-entities to (N)-SAPs and to (N+1)-entities

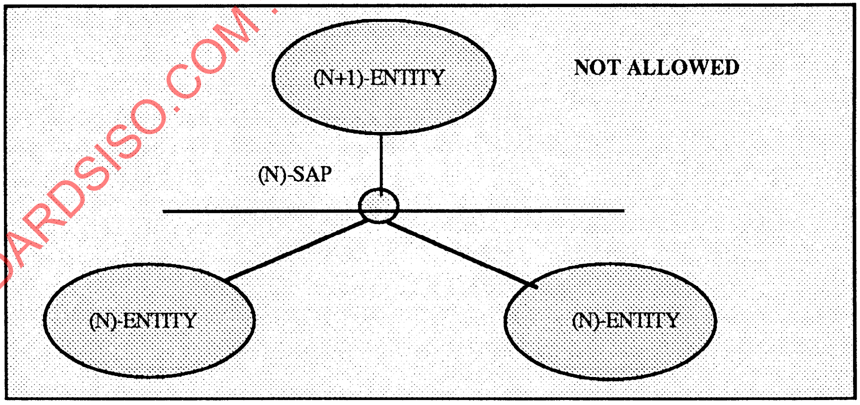


Figure 1b - Not allowed relationships of (N)-entities to (N)-SAPs and to (N+1)-entities

5.2.4 (N)-addresses and (N)-SAP-addresses

An (N)-address identifies a set of (N)-SAPs which are all located at the boundary between an (N)-subsystem and an (N+1)-subsystem. This definition used on e.g. the Network Layer results in the following : a Network-address identifies a set of NSAPs (Network-SAPs). An (N)-address is used to locate an (N+1)-entity or several (N+1)-entities that all provide the same functionalities.

An (N)-SAP-address is an (N)-address which identifies only a single (N)-SAP. Thus there exists a significant difference between an (N)-address and an (N)-SAP-address. There may be functions of a layer which require an (N)-address to identify the single (N)-SAP actually used to support the communication. It is therefore not a property of addresses themselves to determine the course of action taken, but an explicit decision on a layer-by-layer and protocol-by-protocol basis whether a specific (N)-address identifies a single (N)-SAP or a set consisting of more than one (N)-SAP. This decision can also be affected by local consideration related to the configuration of the real open system.

As a consequence of the above clauses, many organisations are possible, for example :

a) multiple (N)-entities which are linked to a single (N+1)-entity ;

b) multiple (N+1)-entities providing the same functionalities which use the services of a single (N)-entity, etc.

5.2.5 The use of (N)-addresses

Naming and addressing mechanisms are an essential aspect of Open Systems Interconnection. Real Open Systems, even while being fully conformant to OSI protocols in all seven layers, may well be unable to set up a dialogue because of inconsistencies among their naming and addressing policies.

Basically, addressing rules must allow an application-entity residing in a real open system to establish an association with a peer application-entity in another real open system. This association implicitly makes use of associations between peer entities established at each of the six lower layers.

The main purpose of (N)-addresses is to make a selection between the various (N+1)-entities available in an (N+1)-subsystem, and thus when the decomposition of an (N+1)-subsystem into (N+1)-entities is complex it is important that the addressing mechanisms allow the addressing scheme attached to that decomposition to remain simple.

ISO 7498-3 states that an (N+1)-subsystem is partitioned into (N+1)-entities for the following reasons :

- to support different (N+1)-protocols or sets of (N+1)-protocols ;
- to accomodate security and/or management requirements ;
- in the case of the application subsystem to distinguish between different application-processes and different application-entities of the same application-process.

ISO 7498-3 states that (N)-addresses are not used

- to distinguish among aspects of protocols that are subject to negotiation (classes, subsets, QOS, protocol versions) or parameter values ;
- to derive routing information above the Network Layer ;
- to distinguish among hardware components.

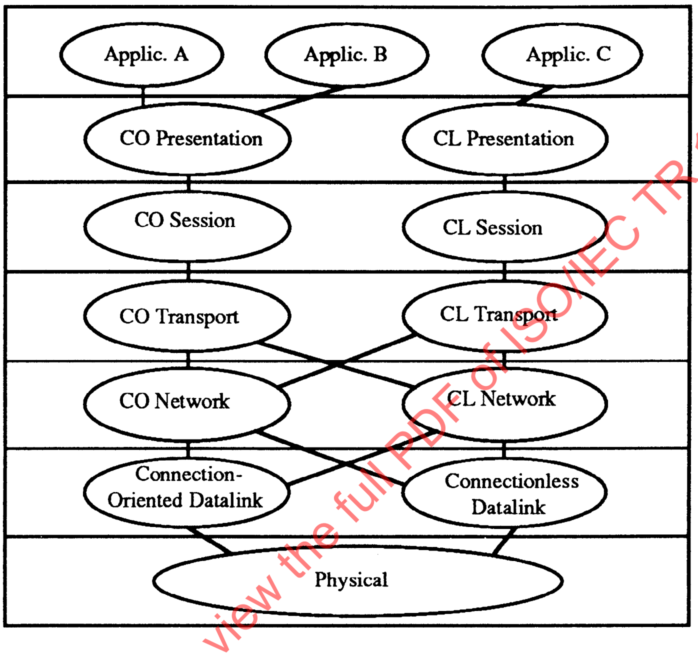
NOTES

1 In some configurations, the normal use of an (N)-address can lead to an (N+1)-entity being wholly contained within a single hardware component. Nevertheless,within the OSIE, the (N)-address identifies the (N+1)-entity; it does not identify the hardware component.

2 As long as different OSI protocols share at least a minimal identification mechanism which allows each type of protocol to be differentiated (e.g. see ISO/IEC TR 9577), they can be considered as being sub-types of a common type and should be handled by one entity. Thus different addresses are not required to discriminate between these different protocols.

These rules are very important because they restrict the degree of complexity of the real open system that may be visible outside this system (i.e. within the OSIE). As mentioned earlier the external view of a real open system must be simple so that it can be easily accessible by other real open systems, even the simplest ones. In order to present a simple external view, it is recommended that a real open system should be modelled as having only one entity in each layer per OSI protocol-type available so that the addressing scheme would be straightforward (see figure 2a).

NOTE - Although examples illustrated by figures 2a to 2d show CO and CL (N)-protocols in separate (N)-entities, it is possible that they may be in the same (N)-entity provided that no ambiguity exists between CO and CL (N)-PDUs.



CO = Connection mode CL = Connectionless mode

Figure 2a - An example of a real open system with an uncomplicated addressing scheme

Now, let us imagine a real open system with a fairly complex internal structure (one possible reason for such a complex and redundant configuration (both in hardware and software terms) could be the need for high reliability). For reasons of simplicity, we shall restrict this example to the Network, Transport and Session Layers (see figure 2b).

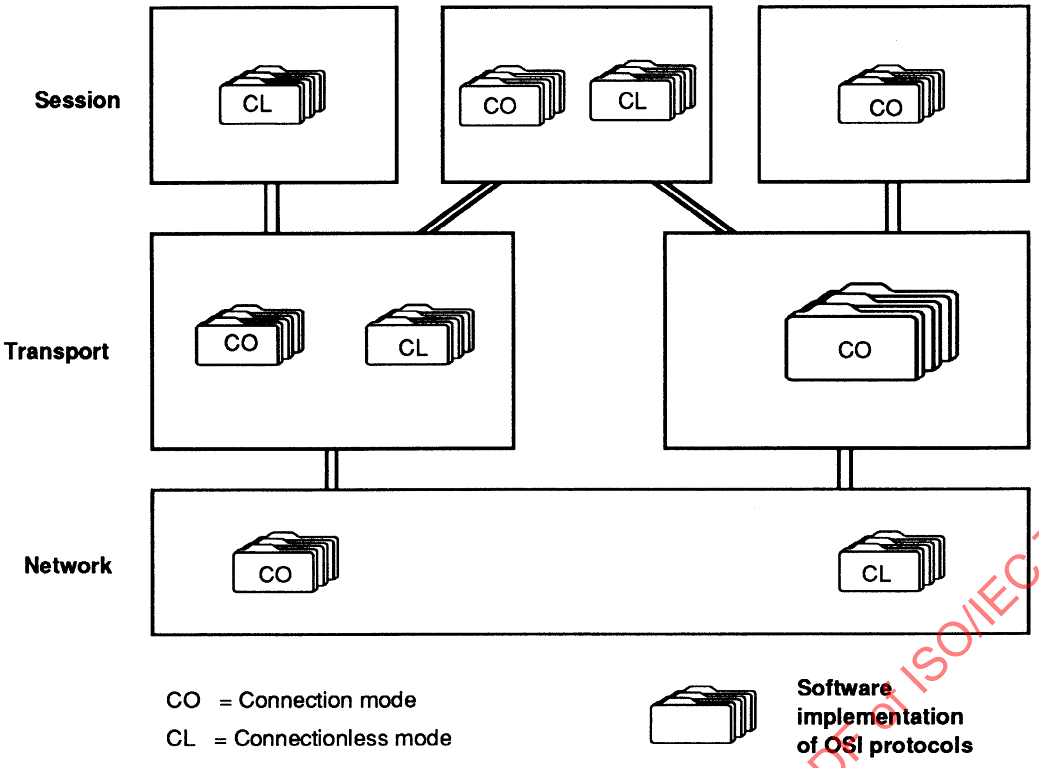


Figure 2b - Example of a complex real internal structure

The structure of this real open system should not be visible from the outside and its addressing scheme should be kept simple. The best way to achieve that is to consider that, at the various layers, there is only one CO-entity and one CL-entity (see figure 2c).

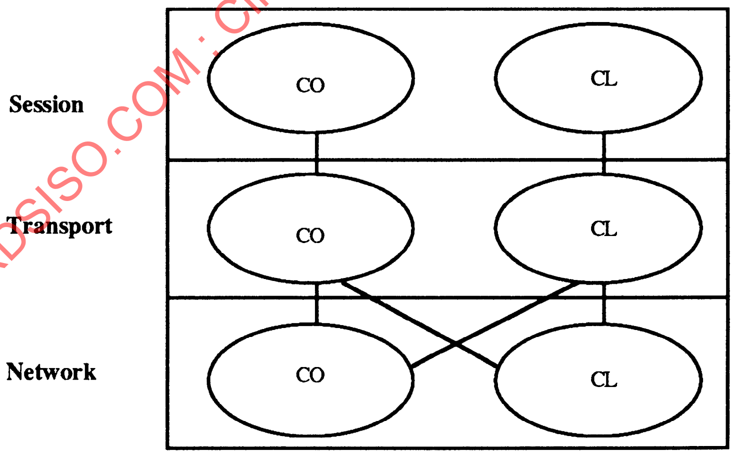


Figure 2c - OSI configuration n°1

Unfortunately, this is not always possible, in particular when there is a need to resume a communication and to get back to the same entity invocation. This may well occurs when the various software implementations are - for some reason - unable to exchange contexts.

In such a case, the OSI configuration of the real open system could look like that illustrated in figure 2d.

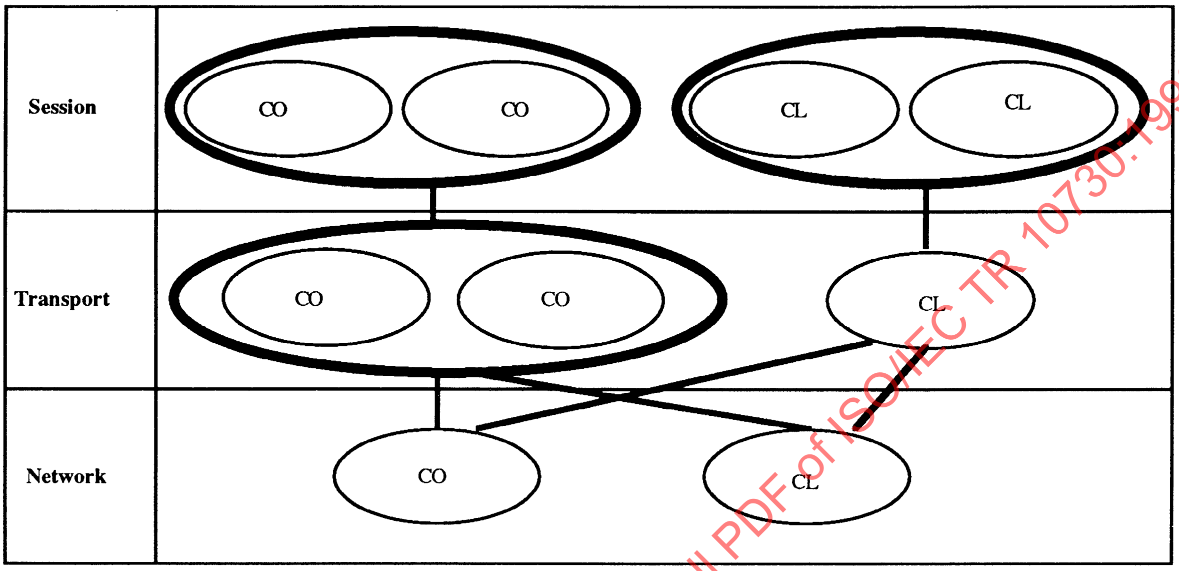


Figure 2d - OSI configuration n°2

Except for the purpose of resuming a communication, there is no reason to distinguish between two CO (or CL) entities in the same (N)-subsystem, and there is no criteria to discriminate among them. It is thus important to be able to access these CO (or CL) entities as if they were a single entity. This can be achieved through the use of an (N)-address encompassing all the various (N)-SAPs leading to these entities. When using these (N)-addresses, the OSI configuration 2 appears equivalent to the OSI configuration 1.

The concept of responding-(N)-address (see 6.2) will allow the recipient system to indicate which specific (N)-SAP is being used for each particular instance of communication.The actual (N)-SAP used, identified by an (N)-SAP-address is provided in the responding-(N)-address.Thus the initiator is notified which particular (N)-SAP is being used for this particular instance of communication amongst the possible set of (N)-SAPs identified by the (N)-address supplied by the initiator. The knowledge of the responding-(N)-address allows the initiator to establish a subsequent connection with the very same (N)-entity (in most cases, the initiator does not care about which entity was selected, nevertheless this mechanism is mandatory when there is a need to resume a communication).

In conclusion, the concepts of (N)-address and responding-(N)-address are essential to the future of OSI, in particular because they allow for the evolution of real open systems towards internal complexity while preserving the external simplicity.

ISO 7498-3 states that each layer standard may, if necessary, impose constraints on the concepts of (N)-addresses and responding-(N)-addresses (e.g. : impose that an (N)-address always be a simple (N)-SAP -address). Such restrictions should be imposed only where their necessity is proven, as they will constrain the flexibility of the addressing concepts.

It must be clearly understood that some existing ISO standards do not make the distinction between (N)-addresses and (N)-SAP-addresses and thus usually state their addressing properties in terms of (N)-SAP-addresses. This is because they were developed before ISO 7498-3 and does not imply that they impose the restriction that (N)-addresses should refer to a single (N)-SAP. On the contrary the properties they state in terms of (N)-SAP-addresses usually naturally extend to (N)-addresses so that a revision of these documents in view of the concepts defined in ISO 7498-3 would be both necessary and easy to handle.

6 Architecture and naming

6.1 Decomposition of names

It is of utmost importance that names be unambiguous within the environment within which they are used. To effect this, registration mechanisms are used, whereby authority is given to some organisation to hold a register of names for objects of some given type, for instance for application entity titles. It is the responsibility of such an authority to ensure unambiguity, i.e. that different objects have different names.

Within the OSIE, there may be a very large number of objects of a given type, and it would be impractical for a single authority to maintain the required register on a worldwide basis. In such a case, it is possible for the registration authority to delegate its operation to subauthorities, provided that a mechanism be clearly instituted that would allow to distinguish between the names registered by the subauthorities. This is usually done by a prefixing rule where the name of an object registered by a subauthority always includes a prefix that identifies in some way the subauthority itself.

Delegation of authority can be repeated as long as needed, until reaching a level of subauthority such that handling of the register becomes reasonable, both in terms of the size of the register itself and in terms of ease of access to the authority for those seeking registration of some object.

One consequence of such a scheme is that names may well become unpleasantly long. This may be an issue of significance when human beings are concerned, but is a trifle when computer communications are concerned.

Another consequence is that the procedure lends itself to the generation of synonyms : the registration of an object may be done, willingly or not, through more than one subauthority, thus leading to the assignment of more than one name to the same object. This does not destroy the fundamental property of name unambiguity, as a given name still designates a single object. But it may create difficulties or awkwardness for some operations : for instance, in the case of directories, which must have as many entries as there can be synonyms for an object, or in the case of some applications which may be in trouble if they fail to recognize the identity of objects which bear different names.

It has however been generally felt that such difficulties are easier to resolve than the avoidance of synonyms by cooperation between subauthorities.

6.2 Addressing information in OSI services

Addressing information is required in OSI services in order to allow an (N+1)-entity to indicate to the (N)-service provider where the peer-(N+1)-entity which is the target of the service exchange is to be found. It follows from the general nature of OSI services that this addressing information is defined only in terms of the semantics assigned to it, not in terms of actual syntactic encoding. Actual encoding will be provided within the protocol elements used to convey the significance of the addressing information to the communication partner.

It is important to distinguish between the semantics of an (N)-address and the syntax used to represent an (N)-address within a given open system ; the semantics of an (N)-address are conveyed to the peer (N)-subsystem ; the syntax of an (N)-address is a local issue and different representations may be used in different open systems. Thus the values of corresponding parameters on the sender and the recipient side must be semantically equivalent, but need not be syntactically identical.

When N is greater than 3 (i.e. above the Network Layer), an (N)-address does not only include addressing information pertinent to the (N)-layer, but also contains addressing information pertinent to the (N-1)-layer. An (N)-address is passed as a parameter of (N)-service primitives through an (N)-SAP. The (N-1)-entity can determine, via the (N-1)-directory-functions (see 6.5), the (N-1)-address from the passed (N)-address.

The addressing information is usually defined in OSI services as parameters of service primitives, needed for each service primitive in the case of the connectionless mode of operation, but only for the set of primitives pertaining to the connection establishment phase in the case of the connection mode of operation.

(N)-addresses are of three types, calling-(N)-address, called-(N)-address and responding-(N)-address.

NOTE - In the service definition of a particular layer, calling-(N)-addresses may be referred to as "source-(N)-addresses" and called-(N)-addresses may be referred to as "destination-(N)-addresses". Throughout ISO 7498-3 however only the preferred terms calling-(N)-address and called-(N)-address are used.

Whenever an (N+1)-entity wishes to establish an (N)-connection or to issue a connectionless unit of data, it must indicate the address(es) of the intended partner(s) and may indicate its own address. The partner address is indicated as the called-(N)-address parameter. This address indicates the (N)-address of the (N)-SAP or set of (N)-SAPs which gives access to the desired partner. In order to obtain this information, the calling-(N+1)-entity makes local usage of directory functions (see 6.5). They may in turn, when needed, require access to an internal or external Directory Facility (see 7.3).

In the called open system, a local mechanism is used to select an appropriate (N)-SAP in the case of several (N)-SAPs sharing the same (N)-address.

The address of the (N)-SAP attached to the initiating (N+1)-entity (i.e. the entity which initiated the connection or issued the connectionless primitive) is conveyed in the request primitive as the calling-(N)-address parameter. This information is provided as a service primitive parameter on the calling side, although it is usually not necessary because it is implicitly known by the calling system. It is carried to the recipient system by the (N)-protocol ; it is made available to the called (N+1)-entity as a parameter of the (N)-service indication primitive in order to provide it with the information needed for identification of the caller and potentially for future recall.

When a called-(N)-address refers to a set of more than one (N)-SAP, it may be useful that the recipient system indicates the address of the specific (N)-SAP which is chosen locally. This is done by use of the service primitive parameter responding-(N)-address. This specific (N)-SAP address is then conveyed back to the caller and provided to it in an (N)-service confirm primitive, so that it allows future recall to the very same (N)-SAP through which the original connection had been established at the recipient. As a consequence, the responding-(N)-address found in the confirm primitive is not necessarily identical to the called-(N)-address initially provided in the (N)-service request primitive.

Thus the responding-(N)-address is a parameter which may appear in an (N)-service response or confirm primitive issued for the purpose of connection establishment but which is not applicable for connectionless mode. It indicates the (N)-address at the (N)-recipient and thus identifies a set of (N)-SAPs at this (N)-recipient.

The fact that the responding-(N)-address identifies a single (N)-SAP (rather than a set of (N)-SAPs) may be a local choice (when the standards specific to the (N)-layer do not impose any constraints) or may be explicitly required by some (N)-layer standard. In practice, it seems to be sensible for a real open system designer to impose such a constraint on responding-(N)-addresses internally, even where it is not mandatory in the applicable standards.

6.3 Addressing Information in OSI protocols

At the protocol level, the addressing information is needed during the establishment (and possibly release) phase in the connection oriented mode of operation and for all data transfers in the connectionless mode of operation. Actually, some sort of addressing does exist during the data transfer phase in the connection oriented mode of operation when multiplexing takes place, but this is limited to referencing of the relevant connection-endpoint-identifiers, and is not seen as a true addressing issue.

NOTE - During the data transfer phase in the connection oriented mode of operation local references are used to discriminate the various connections attached to an (N)-entity. Such local references, the connection-endpoint-identifiers, are assigned during the connection establishment phase. They allow to point to the very entity invocation which, at each participating open system, uses the established connection. These local references remain valid as long as the connection remains in operation. They need not to have the same value at the end-points of the connection.

The addressing information presented in establishment (and possibly release) service primitives in the connection oriented mode of operation and in all primitives in the connectionless mode of operation is carried between open systems as part of the protocol-control-information (PCI) of protocol data units (PDUs). A specific field within the (N)-PCI is used to this effect : the (N)-protocol-addressing-information ((N)-PAI) field. The encoding of this field is part of the specification of the protocol.

The nature of the addressing information carried within the (N)-PAI falls into two categories, full addresses and/or selectors, depending upon the place of the corresponding protocol within the hierarchy of OSI protocols.

At layers 1 and 2, and at layer 3 within subnetworks, where the scope of addressing is restricted to a specific environment within a subnetwork, the notion of a selector is not valid and, as a consequence, the protocol carries the full semantics of the addressing information.

At the Network layer, complete information must be conveyed in the protocol so as to allow unambiguous identification of the parties within the OSI Environment. Because of this requirement for unambiguous identification, network-addresses require registration.

Above the Network Layer, the scope of addressing is again restricted, as the entities to be addressed lie within the open system identified at the network level by use of the network-address. It is possible to take advantage of this fact by using local selectors instead of full addresses and then, if required, make them shorter than full addresses (e.g. to enable potential saving on protocol overhead).

In summary, the information needed to access an application-entity is the tuple

network-address, T-selector, S-selector, P-selector,

where the network-address is carried by network protocols, and T-, S- and P-selectors are carried respectively by transport, session and presentation protocols.

Above the Network Layer, the (N)-address is thus equivalent to the pair [(N-1)-address, (N)-selector]. It should be noted that an (N)-selector value unambiguously identifies a set of (N)-SAPs which are all in the same (N)-subsystem and that an (N-1) selector value cannot be derived from an (N)-selector value.

(N)-selector values are specified locally by each open system for the transport, session and presentation protocols in order to respectively identify session-, presentation- and application- entities. Because (N)-selectors only need to be unambiguous within their respective (N)-subsystems, there is no need for registration authorities. Their values need however to be brought to the knowledge of potential partners through some convenient mechanism, such as bilateral agreement or use of external Directory Facilities.

The abstract syntax of the (N)-selectors is not visible to any other open system ; hence the choice of representation is an entirely local issue. The values of (N)-selectors, the scope of which is unambiguous within the corresponding open system, are chosen by the local administration of an open system ; the local administrator must choose the abstract syntax and encoding technique to be used to represent this value. The mechanism for representing the abstract syntax and encoding technique associated with an (N)-selector value is not within the scope of ISO 7498-3.

The current standards that specify (N)-selectors specify neither an abstract syntax (e.g. ISO 646 characters, CCITT T.61 characters) nor an encoding technique for values in (N)-selectors. The standards simply state that an (N)-selector value is an octet string. For these reasons, it is the actual encoded values that must be made available to other open systems. This can be achieved by recording these actual encoded values in remote Directory Facilities and by ensuring that administrators provide these actual encoded values where remote Directory Facilities are not used.

6.4 Relationship between Addressing Information, OSI-services and OSI-protocols

This topic is illustrated by the use of an example which shows the relationship between the addressing information in OSI services (see 6.2) and the addressing information in OSI protocols (see 6.3) for a presentation-address and the related tuple.

From a presentation address, it is possible to derive the session-address, transport-address and network-address. One possible representation for a presentation-address is shown in figure 3.

Presentation addressing information	Session addressing information	Transport addressing information	Network addressing information

Figure 3 - Possible representation for a presentation-address

This representation describes one possible form of abstract semantics for a presentation-address. The abstract syntax of each of the constituent parts (e.g. session addressing information) is an issue local to that system ; but each constituent part identifies a set of (N)-SAPs at the (N)-subsystem/(N+1)-subsystem boundary.

Another possible representation for the presentation-address is one that does not make visible in the addressing space the boundaries between each (N)-layer. In this case, the (N-1)-address can be derived from the (N)-address using a table lookup mechanism at each subsystem where this representation is used.

The abstract semantics of a presentation-address are conveyed to the peer open system by means of the P-selector, S-selector, T-Selector and network-address, which are conveyed, using some encoding technique, in the presentation-PAI, session-PAI, transport-PAI and network-PAI, respectively.

It can thus be seen that there is a clear relationship between the "presentation addressing information" and the P-selector, the "session addressing information" and the S-selector, the "transport addressing information" and the T-selector, and the "network addressing information" and the network-address. The "(N)-addressing information" and the (N)-selector, which both have to be defined in an abstact syntax, identify the same set of (N)-SAPs. The abstract syntax may or may not be the same for both.

6.5 (N)-directory-functions

ISO 7498-3 defines conceptual (N)-directory-functions. This name ((N)-directory-functions) may be misleading as these functions are not at all related to any access to a Directory Facility (see 7.3) ; they are just abstract concepts which were developed in order to allow for the description of how an (N)-subsystem derives the relation between (N)-addresses, (N-1)-addresses, (N)-selectors, etc...

It is important to note that, being an abstract concept, (N)-directory-functions do not need to be implemented as such.

ISO 7498-3 defines seven initiator (N)-directory-functions and four recipient (N)-directory-functions.

Among these the following are being used in the example of 8.2 :

. The Initiator Addressing Function 2 - IAF2

For this function,

1) the input parameters are : CALLED-(N)-ADDRESS and LOCAL information ;

2) the output is : CALLED-(N-1)-ADDRESS.

. The Initiator Addressing Function 3 - IAF3

For this function,

1) the input parameters are : CALLED-(N-1)-ADDRESS, CALLING-(N)-ADDRESS and LOCAL information ;

2) the output is : CALLING-(N-1)-ADDRESS.

. The Initiator PAI Function 1 - IPF1

For this function,

1) the input parameter is : CALLED-(N)-ADDRESS ;

2) the output is : CALLED-(N)-PAI.

. The Initiator PAI function 2 - IPF2

For this function,

1) the input parameter is : CALLING-(N)-ADDRESS ;

2) the output is : CALLING-(N)-PAI.

. The Recipient Addressing Function 1 - RAF1

For this function,

1) the input parameters are : CALLED-(N)-PAI, CALLED-(N-1)-ADDRESS, and LOCAL information ;

2) the output is : CALLED-(N)-ADDRESS.

. **The Recipient Addressing Function 2 - RAF2**

For this function,

1) the input parameters are : CALLING-(N)-PAI and CALLING-(N-1)-ADDRESS ;

2) the output is : CALLING-(N)-ADDRESS.

. **The Recipient Addressing Function 3 - RAF3**

For this function,

1) the input parameters are : CALLED-(N-1)-ADDRESS and LOCAL information;

2) the output is : RESPONDING-(N-1)-ADDRESS.

6.6 Layer Specific Aspects

Not all (N)-directory-functions apply at a given layer. The general use of these functions is described in the examples of 8.2. Nevertheless, because of the complexity of their internal structure, naming aspects at the Application Layer and at the Network Layer need to be further specified.

6.6.1 The Application Layer

6.6.1.1 Application-Processes and Application-Entities

In a given end system, application-processes are the elements which perform the information processing for particular applications. They are identified by application-process-titles which must be unambiguous throughout the OSIE.

Application processes in different end systems may need to cooperate in order to perform information processing for a particular application. For this purpose they include and make use of communication capabilities which are conceptualised as application-entities. Application-entities are identified by application-entity-titles which must be unambiguous throughout the OSIE. An application-process may make use of communication capabilities through one or more application-entities but an application-entity can belong to only one application-process. Each application-entity is attached to one or more presentation-service-access-points and hence the application-entity-title is associated with the corresponding presentation address(es).

6.6.1.2 Types and invocations

For an application-process (respectively an application-entity), it may at times be desirable to refer to the "type" it belongs to. This is done through the use of application-process-type-titles (respectively application-entity-type-titles) which must be unambiguous throughout the OSIE, and thus must be registered.

On the other hand, it may at times be necessary to make the distinction between the various invocations of a given application-process (respectively application-entity) running concurrently on the open system. This is done through the use of application-process-invocation-identifiers (respectively application-entity-invocation-identifiers) which must be unambiguous within the scope of the application-process (respectively the scope of a pair (application-process-invocation, application-entity)), and thus do not have to be registered.

6.6.1.3 Associations

For communication purposes, application-entity-invocations have to handle one or more application- associations. When needed application-associations are identified by application-association-identifiers.

These identifiers need only to be unambiguous within the scope of the cooperating application-entity-invocations (i.e. the invocations at the endpoints of the association), and thus do not have to be registered.

6.6.1.4 Application Layer naming structure

In order to simplify the construct of names and to minimize the need for registration authorities, titles and identifiers in the Application Layer follow a structuring scheme (see figure 4 for an example) :

a) the application-process-title is the basic element for this structuring scheme. It may be constructed using the system-title as the basis to form its name ;

b) the application-entity-title is composed of an application-process-title plus an application-entity-qualifier ;

c) when it is necessary to identify it, an application-process-invocation is identified by a pair

1) application-process-title of the process to which it belongs, and

2) application-process-invocation-identifier which is unambiguous within the scope of the application-process ;

d) when it is necessary to identify it, an application-entity-invocation is identified by the set

1) application-entity-title of the entity to which it belongs,

2) application-process-invocation-identifier which is unambiguous within the scope of the application process, and

3) application-entity-invocation-identifier which is unambiguous within the scope of both the application-process-invocation and the application-entity.

NOTE - The way that application-entity-titles and application-process-titles are used with Application Directory Facilities to obtain addressing information is described later in 7.3.

The following is an example of a possible naming structure for a “banking application” :

- a banking application provides two facilities, namely a file transfer capability (using FTAM) and a transaction processing capability (using TP) ;

- the banking application could be given the application-process-title "SEOUL-ABC" ;

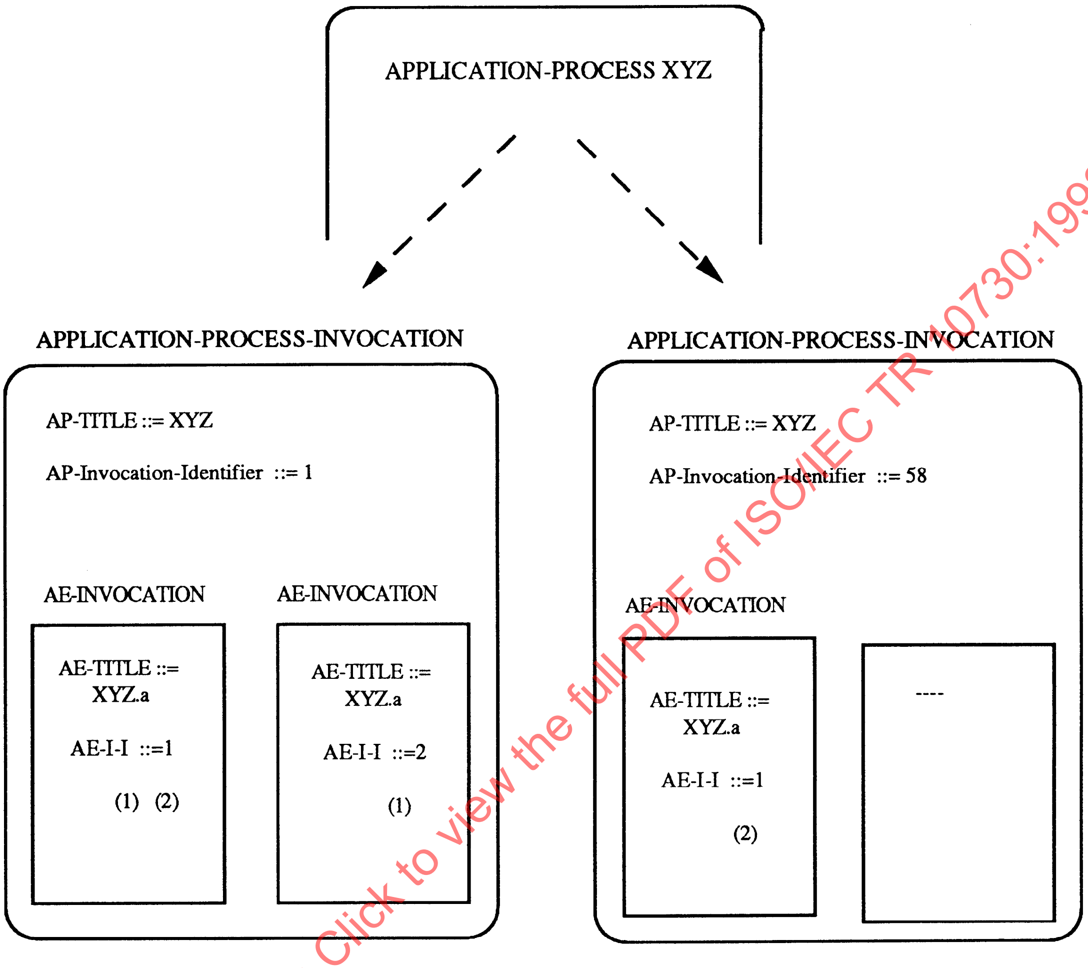
- the FTAM capability could be given the application-entity-type-title of "FTAM". The TP capability could be given the application-entity-type-title of "TP" ;

- the application-process "SEOUL-ABC" could have two application-entities, one of type "FTAM" and one of type "TP" ;

- the application-entity-title of type "FTAM" could be assigned the name "SEOUL-ABC.F". The application-entity-title of type "TP" could be assigned the name "SEOUL-ABC.T";

- the utilisation of the entity "SEOUL-ABC.F" in an association with a remote FTAM-entity "SYDNEY-ABC.F" is an application-entity-invocation. When necessary, this entity-invocation could be identified by the identifier "24" in the application-entity "SEOUL-ABC.F", while the peer entity-invocation could be identified by the identifier "µß".

Figure 4 shows another example of application processes/entities titles and invocation identifiers



NOTES

1 These application-entity-invocation-identifiers must be different as they are associated to application-entity-invocations which belong to the same application-process-invocation.

2 These application-entity-invocation-identifiers may be identical as they are associated to application-entity-invocations which belong to different application-process-invocations.

Figure 4 - Examples of titles and identifiers for application processes and entities

6.6.2 The Network Layer

In the Transport Layer, Session Layer and Presentation Layer the PAI at each layer is only required to convey selectors. In the Network Layer, the PAI must convey the whole network-address.

A network-address is unambiguous throughout the OSIE and identifies a set of NSAPs giving access to a given transport-entity in the same open system. Network-service users cannot derive routeing information from a network-address, nor can they influence the choice of route by means of network-addresses.

In the real world, interconnection among end systems is achieved by using the transmission capabilities of subnetworks. A Subnetwork Point of Attachment (SNPA) is a point of attachment between a real subnetwork and either a real end system or an interworking unit or another real subnetwork.

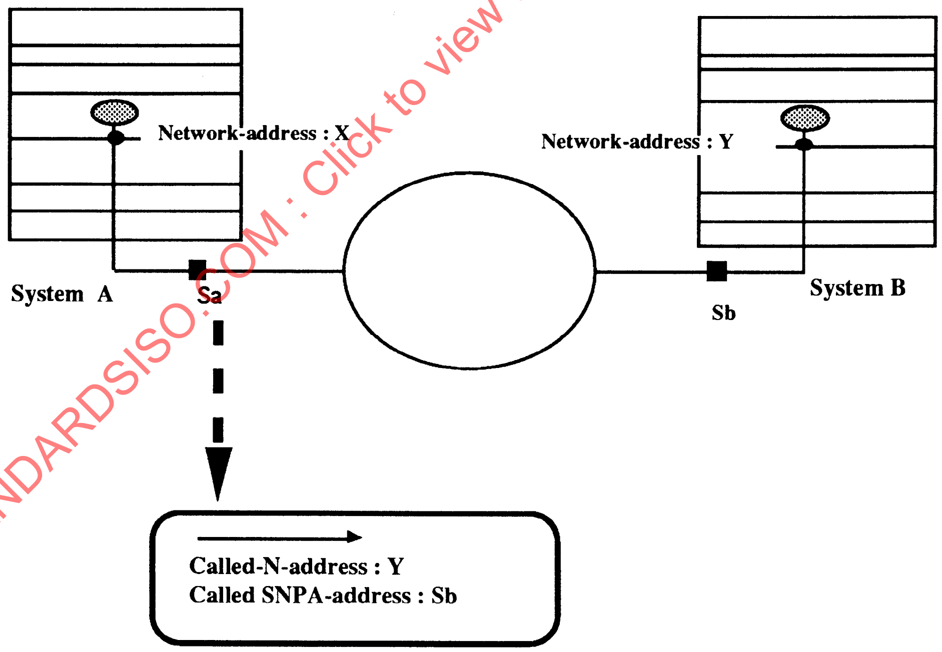
It is a function of the Network Layer to use the network-address for routeing - i.e. to derive an SNPA address from the called-network-address. This SNPA address will be either the SNPA address of the target end system (identified by the called-network-address) or the SNPA address of an intermediate system which is the first (or only) hop in the path to the target end system (the intermediate system then has the task of determining the next applicable SNPA address; this function is repeated until the target end system is reached).

In general, an open system will require use of the Network Address Directory Facility (NADF) to obtain information about network-address - SNPA-address mappings. The relationships between network-addresses and SNPA-addresses may be one-to-one, one-to-many, or many-to-many. The facilities provided by the NADF are likely to be distributed in systems throughout the OSIE. When an open system requires use of these (remotely located) facilities, it will use OSI communication methods to access them.

The network-PAI must convey both the network-address and the SNPA-address. Normally, there will be one field in the network-PAI for conveying the network-address and a separate field in the network-PAI for conveying the SNPA-address.

The network-address in the network-PAI is conveyed transparently between the communicating end systems. The network-PAI field conveying the SNPA-address may contain different SNPA-addresses as it traverses intermediate systems on the path between the two communicating end systems. In fact, at each intermediate system, the Network Layer examines the network-address to determine the next hop in the path to the target end system and places the SNPA-address of the next intermediate system in the SNPA-address field in the network-PAI.

Figures 5 and 6 are examples of the use of network-addresses and SNPA-addresses (for a communication initiated by the end system A, where the end system B is the recipient).



Called-N-address : Y
Called SNPA-address : Sb

Figure 5 - A call between end-systems belonging to the same SNPA-address space

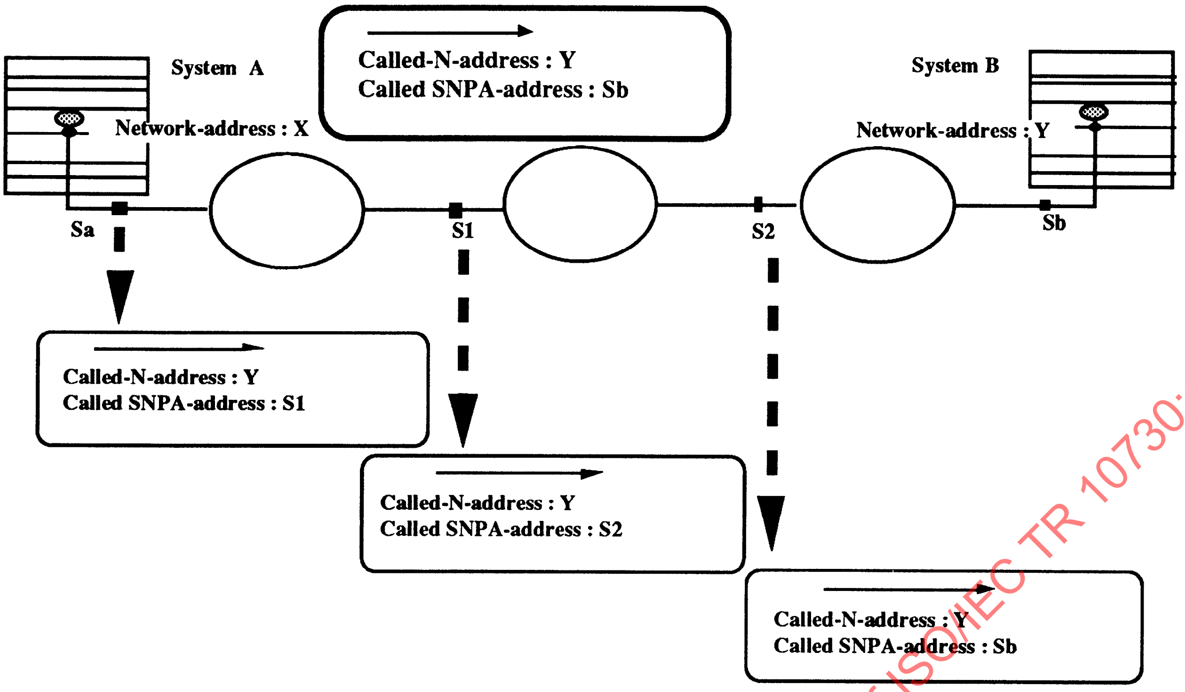


Figure 6 - A call between end-systems belonging to different SNPA-address spaces

NOTE - For the purpose of the examples provided by figures 5 and 6, A and B belong to the same addressing space when system B has an SNPA address that is directly reachable from the initiator system A for any particular instance of communication. An SNPA-address space may cover one or more subnetworks.

7 OSI environment and names

General aspects of naming and addressing have been detailed so far. For real systems and implementations to work properly in the OSIE, names and addresses of information objects have to be defined and made available to some or all communicating entities.

For this purpose, more specific standards have been set up to define how to allocate names or addresses (according to a specific structure or format), or how to store them in a place where they can be accessed when there is a need to do so.

Registration authorities are used for allocation of names and addresses, Directory Facilities are used for storage of names and addresses.

7.1 Naming related standards

As described in clause 1 of this Technical Report, ISO 7498-3 defines the concept of names and addresses, and identifies which objects shall be named in the OSIE.

ISO 7498-3 models (N)-directory-functions (see 6.5 in this Technical Report) and Directory Facilities (see 7.3 in this Technical Report). The purpose of the Directory Facilities is to provide addressing information, given a name. (N)-directory-functions conceptually exist at each layer of the reference model to provide local mappings among addressing information.

ISO 7498-3 also defines the concept of Registration Authority. However, it does not define structures for names or addresses, nor procedures for registration of these names or addresses.

Other standards have been defined to fulfil these more specific needs. The following gives an overview of the current state-of-the-art regarding ISO/IEC JTC1 work in this area. Thus the description of Registration Authorities (see 7.1.3) may be subject to changes as the standards evolve.

7.1.1 The Directory

ISO/IEC 9594 defines a general purpose directory for access by applications, and an associated Directory Service. This standard applies when information has to be stored and retrieved remotely in a Directory entry related to an application process and/or application entity.

It is derived from the directory service originally defined by MHS for its needs. Names of OSI information objects can be stored in, and retreived from the Directory by applications, together with a set of attributes describing the objects (note that the CCITT 1984 version of MHS defined its own directory, while the CCITT 1988 version makes use of the Directory as defined in ISO/IEC 9594).

ISO/IEC 9594 defines the concept of a hierarchical tree known as a Directory Information Tree (DIT). The DIT structure, together with appropriate registration procedures, can be used to ensure that unambiguous names are allocated to objects.

A Relative Distinguished Name (RDN) component is associated with each arc (between two adjacent nodes) in the DIT. An RDN component consists of a set of attribute-value-assertions, where each attribute-value-assertion is made up of two elements, an Attribute Type and an Attribute Value. Syntactically, a Directory name is an ordered sequence of relative distinguished names (RDNs) and thus a Directory name is a descriptive name. So an unambiguous name (i.e a Directory name) is obtained by traversing a path from the root node to a given node and by concatenating (in order) the RDNs of the arcs on the path through the DIT.

NOTE - ISO/IEC 9594 does not provide the Network Address Directory Facility (see 7.3.2).This facility is partly provided by use of routing protocols. Using their own mechanisms, they allow end systems and intermediate systems to exchange information regarding network-addresses, SNPA-addresses and related information such as quality of service parameters and paths availability.

7.1.2 Object Identifiers

ISO/IEC 8824 defines the Object Identifier Tree (OIT). In addition, in Annexes B, C and D of ISO/IEC 8824, the arcs from the root node of the OIT are introduced. This OIT concept together with appropriate registration procedures is used as a source for producing unambigous names for information objects.

The OIT is a hierarchical tree structure with three arcs being specified from the root node. Each node (vertex) of the tree corresponds to an administrative authority responsible for allocating subsequent arcs. Each of these administrative authorities can then delegate responsibility to sub-authorities to allocate arcs beneath them. Each information object is allocated one leaf vertex.

Thus an information object is uniquely and unambiguously identified by the sequence of numeric values labelling the arcs in a path from the root to the vertex allocated to the information object. The resulting object identifier however is a primitive name.

7.1.3 Registration Authorities

ISO/IEC 9834-1 specifies the different phases in the assigment of names.

Registration consists of the assignment of an unambiguous name to an instance of a type of information object.

Registration is carried out by a registration agent. A registration authority is an organization acting as a registration agent. A standard can be a registration agent : this is the case for ISO 8571 in relation to OSI document types which it defines, for instance.

To allow more flexibility, authority can be delegated by the registration agent responsible for a naming domain to subordinate registration agents.

To assure unambiguity, a structured-name-tree approach is taken : a name is constructed in a hierarchical fashion. Thus the name of an information object comprises the name component assigned to it by a registration agent concatenated as a suffix with the name of the agent itself.

ISO/IEC 9834-1 identifies two name forms for use in registration, attribute based form and object-identifier form. These are discussed further in 8.3.

7.2 Registration of names

The procedure for registration allows for a hierarchy of sub-authorities in order to provide convenient management of the registration process.

For instance, some internationally recognized organizations (such as ECMA, EWOS) have requested allocation of an International Code Designator (under ISO 6523), as a first step to be recognised as registration authorities, in order to allocate names for their needs (identification of documents, members, etc).

Nations, identified by a Data Country Code (under ISO 3166), may identify a national organization as registration authority, to allocate names for the needs of organizations. ANSI in the USA, BSI in the UK, AFNOR in France are examples of such registration authorities, for the allocation of organization names.

The subauthorities named under the ICD and DCC nodes may allocate names to various objects. In addition, these subauthorities may further divide their namespace and delegate naming authority over subspaces to subauthorities.

Some name types for which registration authorities allocate names are

- OSI titles :

  . system-titles,
  . application-process-titles,
  . application-process-type-titles,
  . application-entity-type-titles, etc

- MHS addresses (ISO 10021) ;

- Network-addresses (ISO 8348/Add 2).

NOTE - ISO 8348/Add.2 defines the concept of a hierarchical tree for Network addresses. The Network address naming tree, together with the appropriate registration procedures, is used to ensure that unambiguous names are allocated (see 8.3.5).

ISO/IEC 8824, ISO/IEC 9594 and ISO/IEC 9834-1 provide mechanisms to assure unambiguity and global significance of names. The subordinate authorities may define their own rules to structure names in their respective naming domain, provided they comply with these International Standards.

As a result of the distributed nature of the naming process, an information object may have two (or more) structurally different names (synonyms) registered for instance in France and in the UK : not only the values of the sub-nodes are obviously different, but even the number of these sub-nodes, corresponding each to an element of the name, will in general also differ.

7.3 Directory Facilities

Directory Facilities are required to relate names to real locations and real open systems. Directory facilities and related information may be stored locally or remotely, non replicated, partially replicated, or fully replicated.

If Directory Facilities are to be remotely accessed, they have to be accessed through OSI services and protocols.

Two main facilities are defined

a) the ATDF (Application Title Directory Facility) handles application-process-titles and/or application-entity-titles, and returns relevant addressing information ;

b) the NADF (Network Address Directory facility) handles network-addresses, and returns relevant routeing information.

7.3.1 The ATDF

The ATDF (the functionalities of which are illustrated in figure 7) consists of two components in order to return the relevant information associated with the input - i.e. the name - it receives

- a "name resolver" that is able to translate a generic name or an incomplete descriptive name into the associated primitive (non-generic) names or complete descriptive names ;

- a "directory" that returns the addressing information which is associated with a primitive name or a complete descriptive name.

The input to the "name resolver" component of the ATDF is an application-process-title or an application-entity-title. The application-process-title or the application-entity-title may be generic ones (e. g. they may be AP- or AE-type-titles).

If the input is

a) a generic application-process-title, the "name resolver" component of the ATDF returns a list of associated non-generic application-process-titles. These non-generic application-process-titles can then be input to the "name resolver" component of the ATDF (see b-) ;

b) a non-generic application-process-title, the "name resolver" component of the ATDF returns a list of application-entity-titles identifying the application-entities belonging to the considered application-process ;

c) a generic application-entity-title, the "name resolver" component of the ATDF returns a list of associated non-generic application entity-titles.

Any non-generic application-entity-title (which may itself be an output from the "name resolver" component of the ATDF) may be used as input to the "directory" component of the ATDF. The "directory" component of the ATDF is used to obtain the relevant addressing information which is needed to locate the peer entity. This information is defined by a tuple of the form :

(P-Selector,
S-Selector,
T-Selector,
Network-address) (or list of network-addresses)

NOTE - As an N-address refers to a set of NSAPs it is not easy to understand why a list may be returned. A reason for that (and this is the only reason identified so far) is that there may exist several synonymous network-addresses that refer to the same set of NSAPs.

The following provides an example of the use of the ATDF.

An application of title ALPHA within the end system of title '627' requires, as part of its operation, that some computation be made. It happens that this computation requires the use of some high performing parallel processor beyond the capabilities of the processors available within 627.

ALPHA will then want to submit its piece of work to some other end system in which such a processing capability exists, using, for instance, JTM. To this effect, ALPHA must know the title of the application-entity part of that application-process. This can be a specifically identified application, or ALPHA may have no preferred partner and would accept access to any one of a number of such applications. In the former case, ALPHA would contain a primitive name for the title of the remote partner, say "PARPROC84". In the latter case, ALPHA would be provided with a generic application-entity-title for the group of partners, say "PARPROC". This is the case assumed for the continuation of this example.

The second step is to supply ALPHA with the presentation-address of the selected application-entity. Both functions are achieved by actioning the ATDF upon request of ALPHA.
Because the requirement of ALPHA is to communicate using OSI protocols with only one partner, the first step required is to provide it with the information allowing it to choose among the several members of the PARPROC group on the basis of some criteria private to ALPHA.

In this example, the ATDF is assumed to be distributed with a part included within system 627. Then communications between ALPHA and the local part of the ATDF uses private means. When the information about PARPROC resides within that part of the ATDF which is within system 627, no external visibility of the procedure is required, hence no standardized protocols are used.

When the local portion of the ATDF cannot provide the information about PARPROC, it communicates using directory protocols with the remainder of the distributed ATDF, obtains the information on PARPROC, and conveys it to ALPHA through the private local mechanism. Depending upon implementation choices, the local ATDF may decide to take advantage of the information thus gained about PARPROC to store it in case future requirements for the same partner would arise in system 627.

Figure 7 illustrates the various possible inputs and entry points to the ATDF.

7.3.2 The NADF

A calling-network-address and a called-network-address are passed (as parameters of a network-service primitive) by an initiating transport entity to the related initiating network entity. The called-network-address (which is part of the address tuple provided by the ATDF) implicitly identifies the transport entity to be reached on the remote end system. There is thus a need to access this remote end system. The role of the NADF is to provide the information necessary to achieve that.

The NADF requires as input the calling- and the called-network-addresses and returns, using appropriate procedures, the calling- and the called-data-link-address, the called-network-PAI, the called-subnetwork-address-information, the calling-network-PAI and the calling-subnetwork-PAI.

The NADF is different from the ATDF in that the information provided by the NADF is dependant on the relative location of the real open systems which intend to communicate whereas the information provided by the ATDF is independant from this location. Thus when an application-entity-title is input to an ATDF, the same addressing information is always the result (at least for as long as this information is not updated in the ATDF). However, when the initiating end system inputs network-addresses to the NADF, the information that is returned includes routeing information in the form of the first (or only) hop in the path to the target end-system. As described earlier (6.6.2) the SNPA-address of the first hop may be the SNPA-address of the target end system if this system is directly reachable or it will be the SNPA-address of an intermediate system if the target end system is not directly reachable. Thus different results may be obtained from the NADF depending on the relative locations of the two end systems.

The intermediate system(s) (if any) in the path to the target end system will use the NADF in order to determine the next hop SNPA-address and the other addressing information required. The result will be different from the result obtained by the initiating end system.

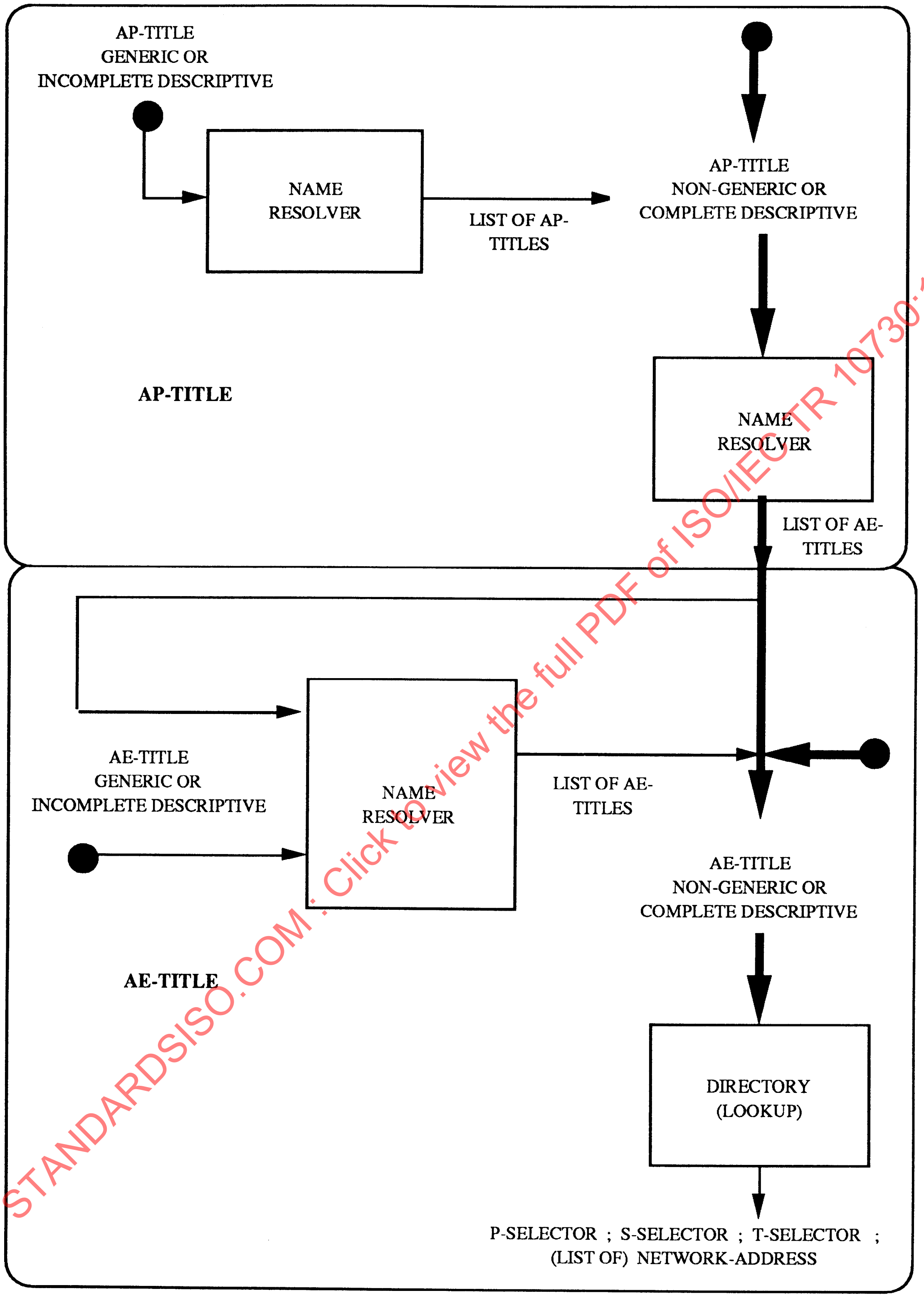


Figure 7 - The ATDF functionalities

8 Examples

8.1 Entities and addresses

It is important to note that, although the internal addressing structure of a real open system is a local matter, its visibility from the outside (i.e. the tuple [Network-address, T-selector, S-selector, P-selector] provided through the means of a Directory Facility or through other means) has to be consistent, so that it allows for the establishment of a connection with the desired application-entity.

The two following examples (figures 8a and 8b) are aimed at clarifying this requirement :

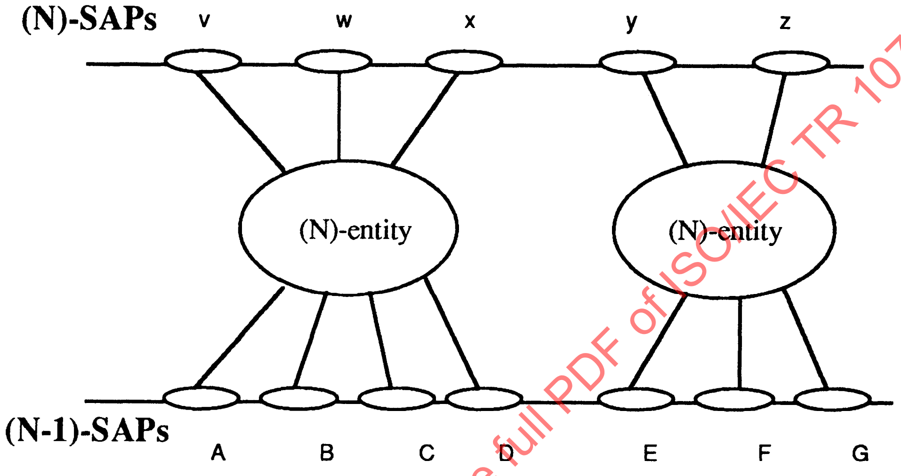


Figure 8a - Relations between selectors associated to (N)-SAPs and (N-1)-SAPs

A,B, ... G are (N-1)-addresses
v,w, ...z are (N)-selectors

The following are possible valid (N)-addresses :

Av, Aw, Ax, Bv, Bw, Bx, Ey, Ez, Gy, Gz

The following are possible invalid (N)-addresses :

Ay, Bz, Fw, Gx

NOTE - The format of (N)-addresses used in this example may appear to imply a hierarchical structuring of an (N)-address from an (N-1)-address and an (N)-selector. This format is only used for simplification. (N)-addresses are purely conceptual and their actual representation in a real open system is a local matter.

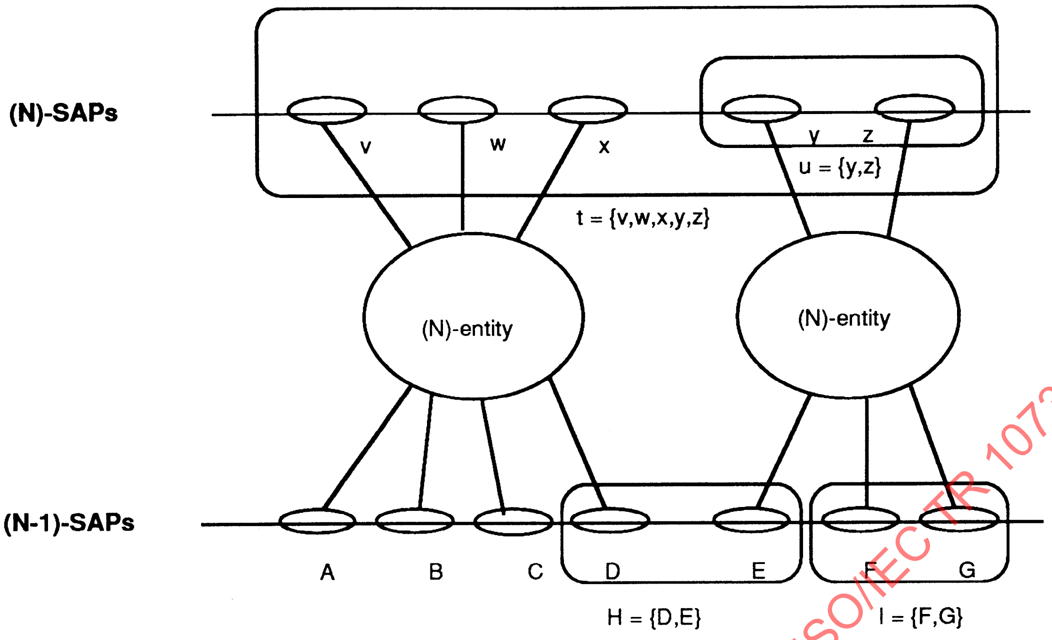


Figure 8b - Relations between selectors associated to sets of (N)-SAPs and (N-1)-SAPs

In figure 8b, Iu, It, Ht are possible valid addresses because for any choice of (N-1)-SAP, there is always an (N)-SAP available. Hu is an invalid address because if (N-1)-SAP D is used for a specific communication, there is no available (N)-SAP.

The example illustrated in figure 9 hereafter shows layers 2 to 5 of a system "S" where the Session Layer includes two connection-mode session-entities respectively called Entity A and Entity B (possible reasons for such an internal design are discussed in 5.2.5). However, from any other open system (and thus from the OSIE) the distinction between A and B does not have to be made as they both provide the same service. A and B are thus accessed as a whole through the use of a single transport-address T1, and the external view of the subsystem is a very simple one.

If, for any reason, a remote system needs, in a subsequent instance of transport communication, to go back to the very same session entity (i.e the session entity which had been using the previous transport connection), it has to be able to make the distinction between the various TSAPs giving access to A and B. This is possible through the use of the "responding-transport-address" which had been provided by S.

Therefore for general communication purposes, remote open systems will not make the distinction between entities A and B and the only transport-address known to them for the connection-mode session-entity will be T1. When a connection is established "S" will return, as a responding-transport-address, the address of the specific TSAP which was used for this instance of communication (e.g. T4). The initiating subsystem may then, if it feels it necessary, keep this transport-address (T4) for use in a subsequent communication. Using T4 instead of T1 would then mean that the remote end system wishes to go back to the very same session-entity with which it had the previous communication.

NOTES

1 For the purpose of simplicity, only transport-address T1 has been decomposed into several TSAP addresses. This decomposition may apply of course to any (N)-address at the discretion of the local administration of system "S".

2 Although the previous example is architecturally valid, the transport service (as currently defined) does not allow the responding-transport-address to be different from the called-transport-address so that the scenario described in this example is not currently possible. This restriction may be removed in a future version of the transport service in order to fully utilise the flexibility provided by the responding-address mechanism.

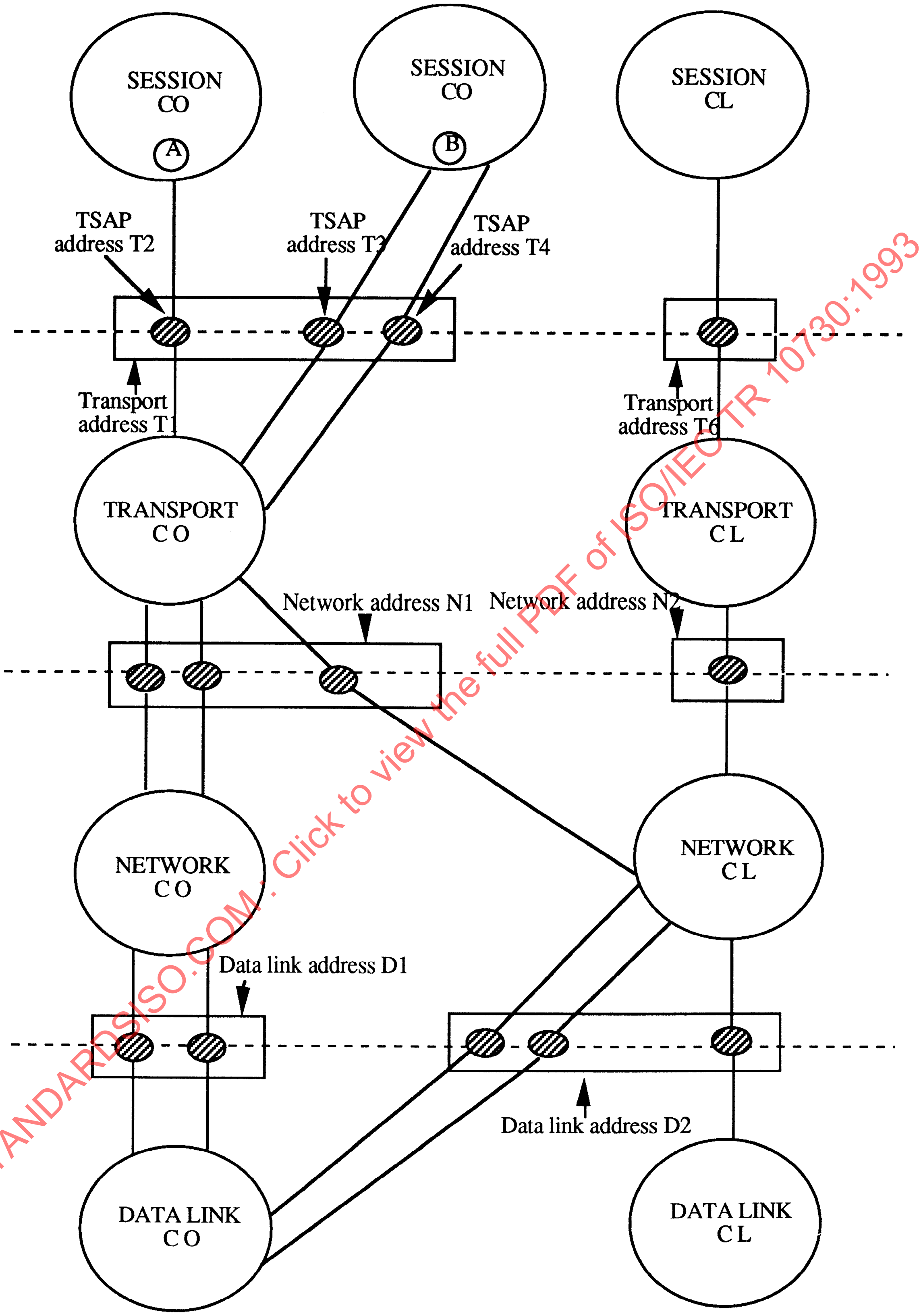


Figure 9 - Example of an addressing structure for layers 2 to 5

8.2 Application-association establishment

8.2.1 Obtaining names

Let us suppose that an application-entity named NAMEA would like to initiate an association with another remote application-entity. Of course, NAMEA must know the name of the remote entity to talk to, an application-entity-title, say NAMEB.

The initiating application-entity sends NAMEB to the ATDF ; the ATDF facility will return a tuple to locate NAMEB, for instance

aofNAMEB = pp ss tt XYZ,

where

aofNAMEB is unambiguous within the OSIE and locates NAMEB ;
pp, ss, tt are names local to the remote system ;
pp is a Presentation selector ;
ss is a Session selector ;
tt is a Transport selector and ;
XYZ is unambiguous within the OSIE and is the network-address of the real system where NAMEB is located.

The selectors are used in the layers from 6 to 4 and conveyed to the corresponding peer entities, within the PAI.

At layer 3, the network entity receives the XYZ name. The network-entity then calls the NADF facility ; these functions return an SNPA-address which refers to the next hop on the path to be followed to reach the transport-entity corresponding to the XYZ address.

NOTE - In the case where a real open system provides both the ATDF and the NADF, the retrieval of information from both Directory Facilities may be made in a single inquiry.

8.2.2 Layered mechanisms in the calling system

Figure 10 shows the main effects of the initiator (N)-directory-functions (see 6.5) in the calling open system when the called- and calling-presentation-addresses are passed to the presentation entity as parameters of a presentation service primitive.

8.2.3 Layered mechanisms in the called system

Figure 11 shows the main effects of the recipient (N)-directory-functions (see 6.5) in the called open system when the network PDU sent by the initiating system and conveying the called- and calling-network-addresses reaches the recipient system.

NOTES

1 Above the Network Layer boundary, only selectors are exchanged between subsystems, and the actual structure of called-, calling- and responding-addresses is a matter purely internal to each specific implementation.

2 The structure used in figures 10 and 11 for (N)-addresses is just an example of a possible internal structure. It has been used because it makes visible the relationship between (N)-selectors and (N)-addresses. Other internal structures may be used for existing implementations.