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

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 Technical reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful .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.

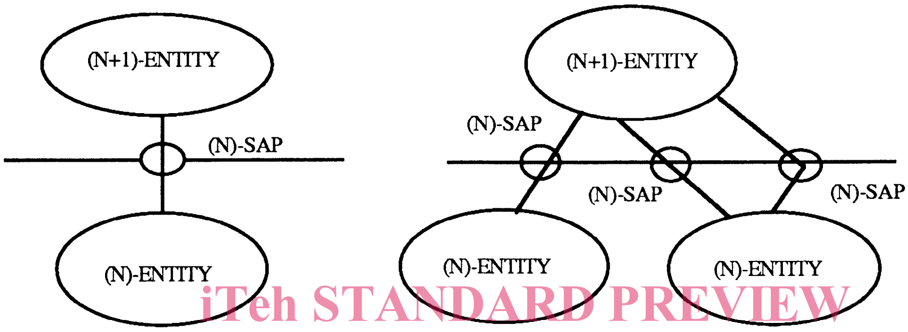


Figure 1a - Allowed relationships of (N)-entities to (N)-SAPs and to (N+1)-entities

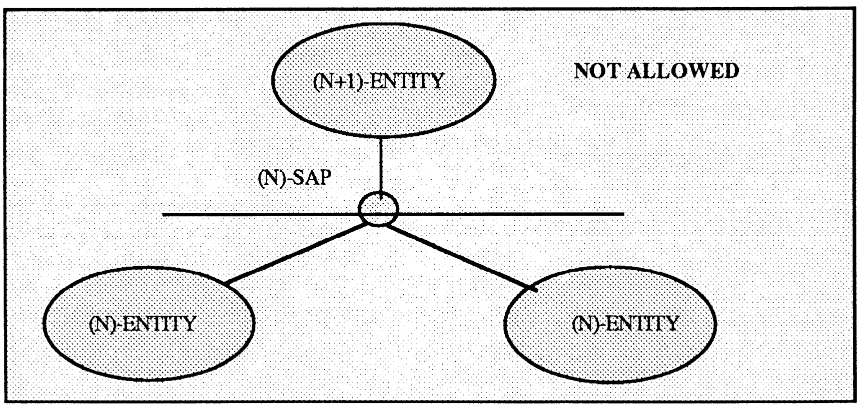


Figure 1b - Not allowed relationships of (N)-entities to (N)-SAPs and to (N+1)-entities

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.

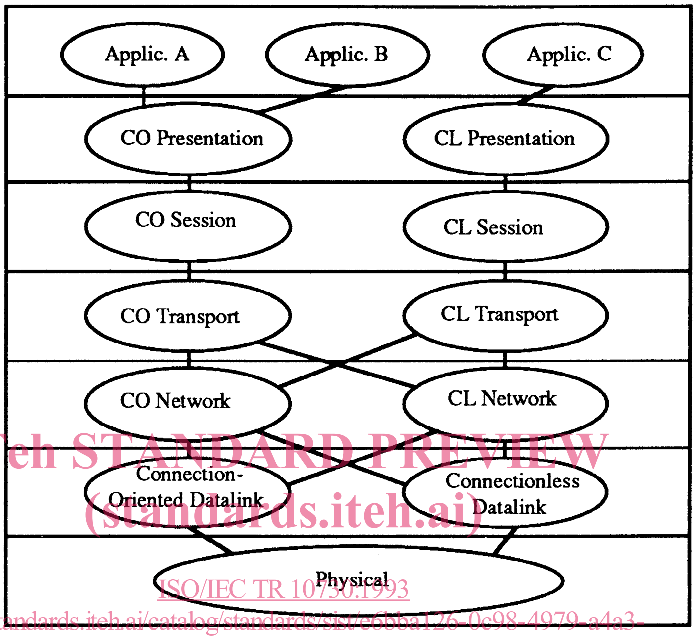
NOTES

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).

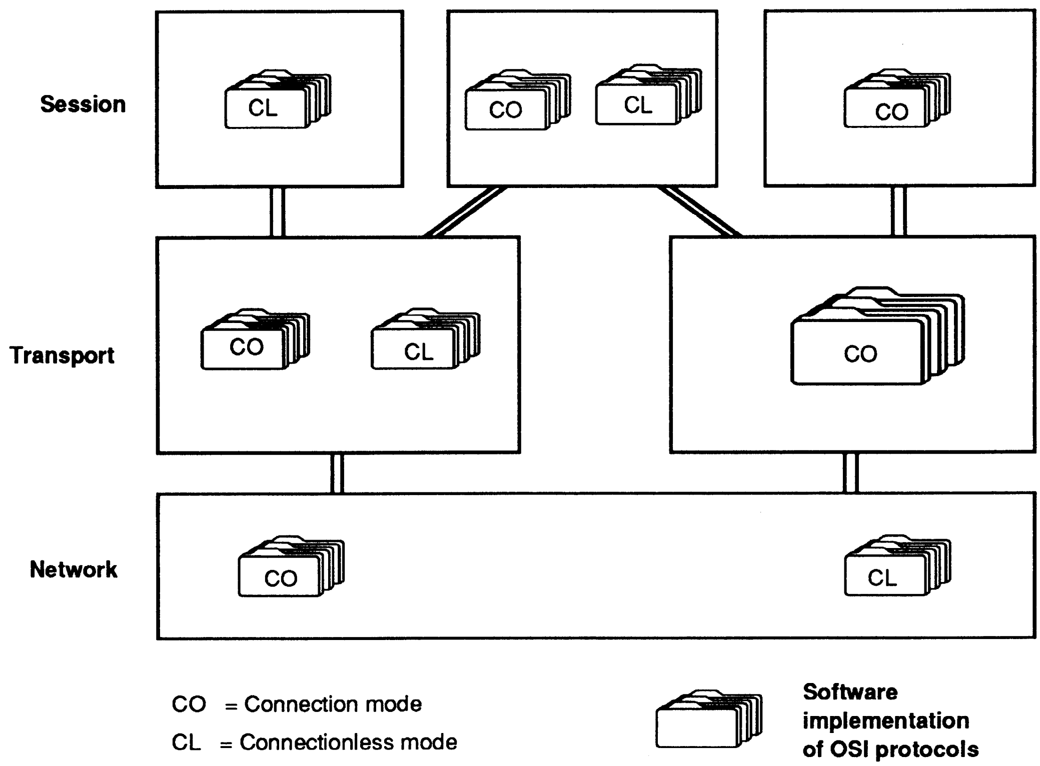


Figure 2b - Example of a complex real internal structure

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).

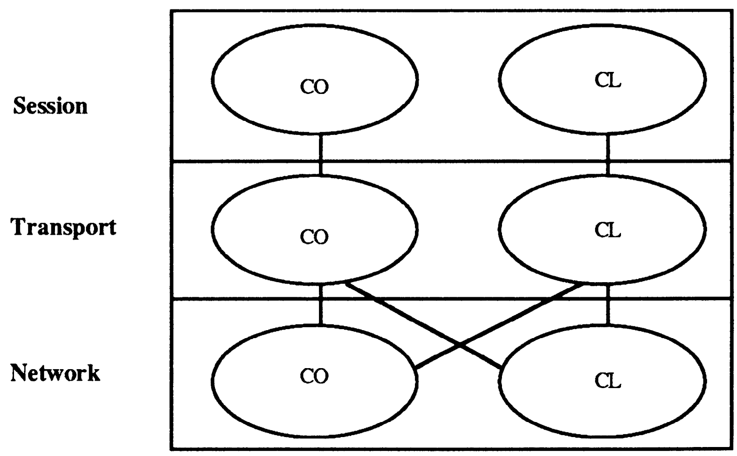


Figure 2c - OSI configuration n°1