# OSI Reference Model—The ISO Model of Architecture for Open Systems Interconnection

HUBERT ZIMMERMANN

*(Invited Paper)*

***Abstract*—Considering the urgency of the need for standards which would allow constitution of heterogeneous computer networks, ISO created a new subcommittee for "Open Systems Interconnection" (ISO/TC97/SC16) in 1977. The first priority of subcommittee 16 was to develop an architecture for open systems interconnection which could serve as a framework for the definition of standard protocols. As a result of 18 months of studies and discussions, SC16 adopted a layered architecture comprising seven layers (Physical, Data Link, Network, Transport, Session, Presentation, and Application). In July 1979 the specifications of this architecture, established by SC16, were passed under the name of "OSI Reference Model" to Technical Committee 97 "Data Processing" along with recommendations to start officially, on this basis, a set of protocols standardization projects to cover the most urgent needs. These recommendations were adopted by TC97 at the end of 1979 as the basis for the following development of standards for Open Systems Interconnection within ISO. The OSI Reference Model was also recognized by CCITT Rapporteur's Group on "Layered Model for Public Data Network Services."**

**This paper presents the model of architecture for Open Systems Interconnection developed by SC16. Some indications are also given on the initial set of protocols which will likely be developed in this OSI Reference Model.**

## I. INTRODUCTION

IN 1977, the International Organization for Standardization (ISO) recognized the special and urgent need for standards for heterogeneous informatic networks and decided to create a new subcommittee (SC16) for "Open Systems Interconnection."

The initial development of computer networks had been fostered by experimental networks such as ARPANET [1] or CYCLADES [2], immediately followed by computer manufacturers [3], [4]. While experimental networks were conceived as heterogeneous from the very beginning, each manufacturer developed his own set of conventions for interconnecting his own equipments, referring to these as his "network architecture."

The universal need for interconnecting systems from different manufacturers rapidly became apparent [5], leading ISO to decide for the creation of SC16 with the objective to come up with standards required for "Open Systems Interconnection." The term "open" was chosen to emphasize the fact that by conforming to those international standards, a system will be open to all other systems obeying the same standards throughout the world.

The first meeting of SC16 was held in March 1978, and initial discussions revealed [6] that a consensus could be reached rapidly on a layered architecture which would satisfy most requirements of Open Systems Interconnection with the capacity of being expanded later to meet new requirements. SC16 decided to give the highest priority to the development of a standard Model of Architecture which would constitute the framework for the development of standard protocols. After less than 18 months of discussions, this task was completed, and the ISO Model of Architecture called the Reference Model of Open Systems Interconnection [7] was transmitted by SC16 to its parent Technical Committee on "Data Processing" (TC97) along with recommendations to officially start a number of projects for developing on this basis an initial set of standard protocols for Open Systems Interconnection. These recommendations were adopted by TC97 at the end of 1979 as the basis for following development of standards for Open Systems Interconnection within ISO. The OSI Reference Model was also recognized by CCITT Rapporteur's Group on Public Data Network Services.

The present paper describes the OSI Architecture Model as it has been transmitted to TC97. Sections II–V introduce concepts of a layered architecture, along with the associated vocabulary defined by SC16. Specific use of those concepts in the OSI seven layers architecture are then presented in Section VI. Finally, some indications on the likely development of OSI standard protocols are given in Section VII.

### *Note on an "Interconnection Architecture"*

The basic objective of SC16 is to standardize the rules of interaction between interconnected systems. Thus, only the external behavior of Open Systems must conform to OSI Architecture, while the internal organization and functioning of each individual Open System is out of the scope of OSI standards since these are not visible from other systems with which it is interconnected [8].

It should be noted that the same principle of restricted visibility is used in any manufacturer's network architecture in order to permit interconnection of systems with different structures within the same network.

These considerations lead SC16 to prefer the term of "Open Systems Interconnection Architecture" (OSIA) to the term of "Open Systems Architecture" which had been used previously and was felt to be possibly misleading. However, for unclear reasons, SC16 finally selected the title "Reference Model of Open Systems Interconnection" to refer to this Interconnection Architecture.

Manuscript received August 5, 1979; revised January 16, 1980.
The author is with IRIA/Laboria, Rocquencourt, France.

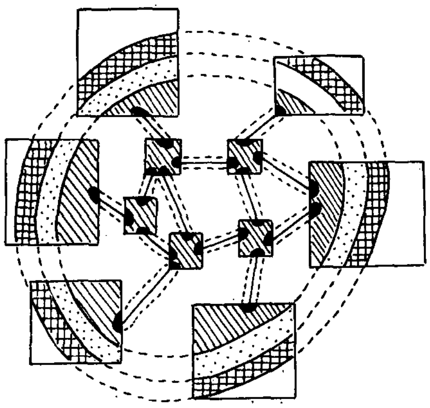


Fig. 1. Network layering.

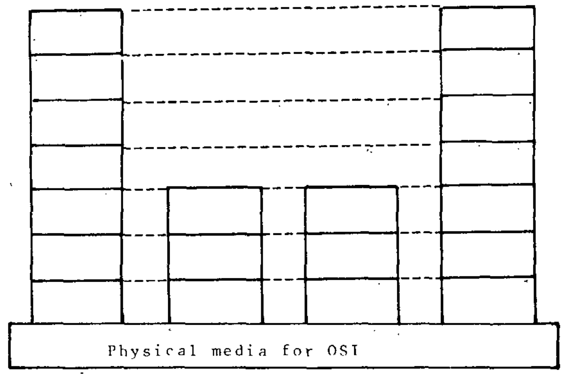



Fig. 2. An example of OSI representation of layering.

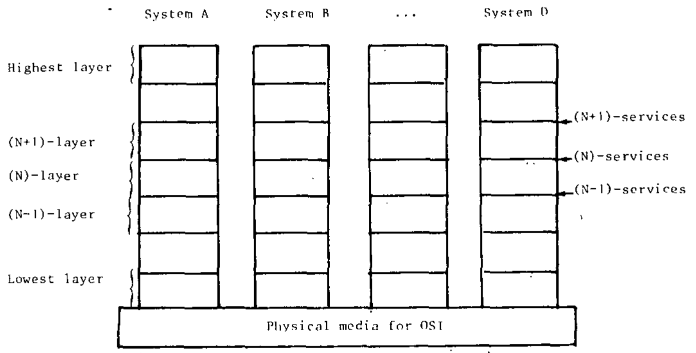



Fig. 3. Systems, layers, and services.

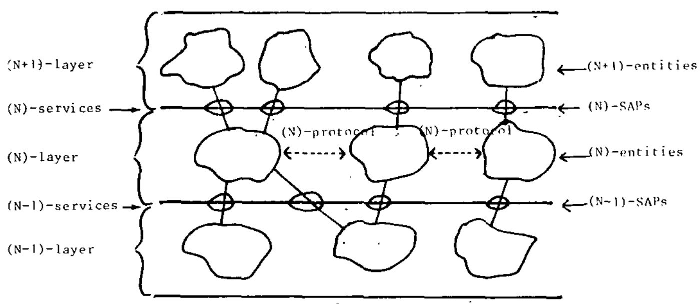



Fig. 4. Entities, service access points (SAP's), and protocols.

## II. GENERAL PRINCIPLES OF LAYERING

Layering is a structuring technique which permits the network of Open Systems to be viewed as logically composed of a succession of layers, each wrapping the lower layers and isolating them from the higher layers, as exemplified in Fig. 1.

An alternative but equivalent illustration of layering, used in particular by SC16 is given in Fig. 2 where successive layers are represented in a vertical sequence, with the physical media for Open Systems Interconnection at the bottom.

Each individual system itself is viewed as being logically composed of a succession of subsystems, each corresponding to the intersection of the system with a layer. In other words, a layer is viewed as being logically composed of subsystems of the same rank of all interconnected systems. Each subsystem is, in turn, viewed as being made of one or several entities. In other words, each layer is made of entities, each of which belongs to one system. Entities in the same layer are termed *peer* entities.

For simplicity, any layer is referred to as the $(N)$ *layer*, while its next lower and next higher layers are referred to as the $(N - 1)$ layer and the $(N + 1)$ layer, respectively. The same notation is used to designate all concepts relating to layers, e.g., entities in the $(N)$ layer are termed $(N)$ *entities*, as illustrated in Figs. 3 and 4.

The basic idea of layering is that each layer adds value to services provided by the set of lower layers in such a way that the highest layer is offered the set of services needed to run distributed applications. Layering thus divides the total problem into smaller pieces. Another basic principle of layering is to ensure independence of each layer by defining services provided by a layer to the next higher layer, independent of how these services are performed. This permits changes to be made in the way a layer or a set of layers operate, provided they still offer the same service to the next higher layer. (A more comprehensive list of criteria for layering is given in Section VI.) This technique is similar to the one used in structured programming where only the functions performed by a module (and not its internal functioning) are known by its users.

Except for the highest layer which operates for its own purpose, $(N)$ entities distributed among the interconnected Open Systems work collectively to provide the $(N)$ *service* to $(N + 1)$ entities as illustrated in Fig. 4. In other words, the $(N)$ entities add value to the $(N - 1)$ service they get from the $(N - 1)$ layer and offer this value-added service, i.e., the $(N)$ service to the $(N + 1)$ entities.

Communication between the $(N + 1)$ entities make exclusive use of the $(N)$ services. In particular, direct communication between the $(N + 1)$ entities in the same system, e.g., for sharing resources, is not visible from outside of the system and thus is not covered by the OSI Architecture. Entities in the lowest layer communicate through the Physical Media for OSI, which could be considered as forming the $(O)$ layer of the OSI Architecture. Cooperation between the $(N)$ entities is ruled by the $(N)$ *protocols* which precisely define how the $(N)$ entities work together using the $(N - 1)$ services to perform the $(N)$ functions which add value to the $(N - 1)$ service in order to offer the $(N)$ service to the $(N + 1)$ entities.

The $(N)$ services are offered to the $(N + 1)$ entities at the $(N)$ *service access points*, or $(N)$ *SAP's* for short, which represent the logical interfaces between the $(N)$ entities and the $(N + 1)$ entities. An $(N)$ SAP can be served by only one

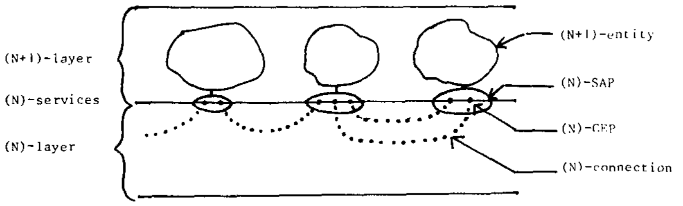



Fig. 5. Connections and connection endpoints (CEP's).

(*N*) entity and used by only one (*N* + 1) entity, but one (*N*) entity can serve several (*N*) SAP's and one (*N* + 1) entity can use several (*N*) SAP's.

A common service offered by all layers consists of providing associations between peer SAP's which can be used in particular to transfer data (it can, for instance, also be used to synchronize the served entities participating in the association). More precisely (see Fig. 5), the (*N*) layer offers (*N*) *connections* between (*N*) SAP's as part of the (*N*) services. The most usual type of connection is the *point-to-point* connection, but there are also *multiendpoint* connections which correspond to multiple associations between entities (e.g., broadcast communication). The end of an (*N*) connection at an (*N*) SAP is called an (*N*) *connection endpoint* or (*N*) *CEP* for short. Several connections may coexist between the same pair (or *n*-tuple) of SAP's.

*Note*: In the following, for the sake of simplicity, we will consider only point-to-point connections.

## III. IDENTIFIERS

Objects within a layer or at the boundary between adjacent layers need to be uniquely identifiable, e.g., in order to establish a connection between two SAP's, one must be able to identify them uniquely. The OSI Architecture defines identifiers for entities, SAP's, and connections as well as relations between these identifiers, as briefly outlined below.

Each (*N*) entity is identified with a *global title*[1] which is unique and identifies the same (*N*) entity from anywhere in the network of Open Systems. Within more limited *domains*, an (*N*) entity can be identified with a *local title* which uniquely identifies the (*N*) entity only in the domain. For instance, within the domain corresponding to the (*N*) layer, (*N*) entities are identified with (*N*) *global titles* which are unique within the (*N*) layer.

Each (*N*) SAP is identified with an (*N*) *address* which uniquely identifies the (*N*)-SAP at the boundary between the (*N*) layer and the (*N* + 1) layer.

The concepts of titles and addresses are illustrated in Fig. 6.

Binding between (*N*) entities and the (*N* – 1) SAP's they use (i.e., SAP's through which they can access each other and communicate) are translated into the concept of (*N*) directory which indicates correspondence between global titles of (*N*) entities and (*N*) addresses through which they can be reached, as illustrated in Fig. 7.

[1] The term "title" has been preferred to the term "name" which is viewed as bearing a more general meaning. A title is equivalent to an entity name.

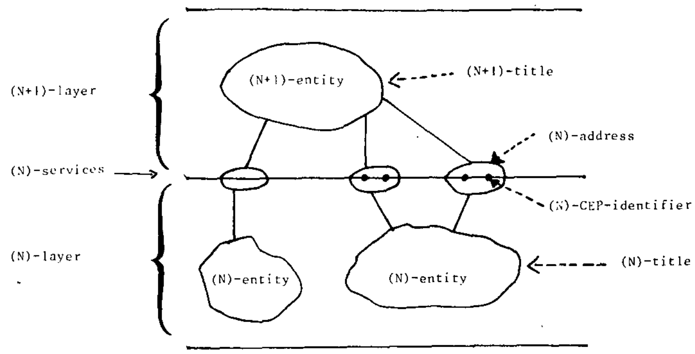



Fig. 6. Titles, addresses, and CEP-identifiers.

| (N)-title | (N-1)-address |
|---|---|
| A | 352 |
| B | 237 |
| B | 015 |
| C | 015 |
| | |

Fig. 7. Example of an (*N*)-directory.

Correspondence between (*N*) addresses served by an (*N*) entity and the (*N* – 1) addresses used for this purpose is performed by an (*N*) mapping function. In addition to the simplest case of one-to-one mapping, mapping may, in particular, be hierarchical with the (*N*) address being made of an (*N* – 1) address and an (*N*) suffix. Mapping may also be performed "by table." Those three types of mapping are illustrated in Fig. 8.

Each (*N*) CEP is uniquely identified within its (*N*) SAP by an (*N*) *CEP identifier* which is used by the (*N*) entity and the (*N* + 1) entity on both sides of the (*N*) SAP to identify the (*N*) connection as illustrated in Fig. 6. This is necessary since several (*N*) connections may end at the same (*N*) SAP.

## IV. OPERATION OF CONNECTIONS

### *A. Establishment and Release*

When an (*N* + 1) entity requests the establishment of an (*N*) connection from one of the (*N*) SAP's it uses to another (*N*) SAP, it must provide at the local (*N*) SAP the (*N*) address of the distant (*N*) SAP. When the (*N*) connection is established, both the (*N* + 1) entity and the (*N*) entity will use the (*N*)CEP identifier to designate the (*N*) connection.

(*N*) connections may be established and released dynamically on top of (*N* – 1) connections. Establishment of an (*N*) connection implies the availability of an (*N* – 1) connection between the two entities. If not available, the (*N* – 1) connection must be established. This requires the availability of an (*N* – 2) connection. The same consideration applies downwards until an available connection is encountered.

In some cases, the (*N*) connection may be established simultaneously with its supporting (*N* – 1) connection provided

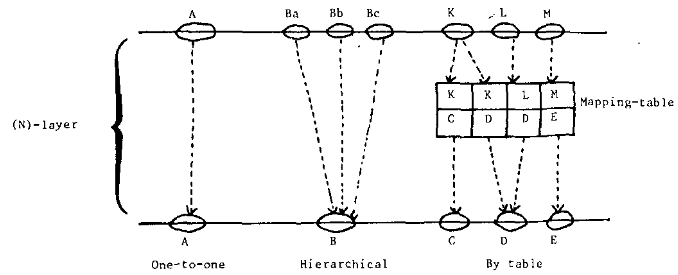



Fig. 8. Mapping between addresses.

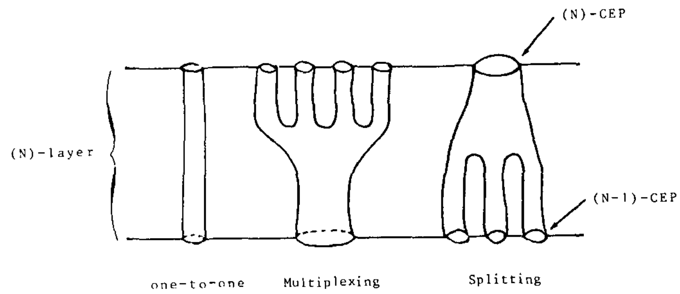



Fig. 9. Correspondence between connections.

the (N − 1) connection establishment service permits (N) entities to exchange information necessary to establish the (N) connection.

### B. Multiplexing and Splitting

Three particular types of construction of (N) connections on top of (N − 1) connections are distinguished.

1) One-to-one correspondence where each (N) connection is built on one (N − 1) connection.

2) Multiplexing (referred to as "upward multiplexing" in [7]) where several (N) connections are multiplexed on one single (N − 1) connection.

3) Splitting (referred to as "downward multiplexing" in [7]) where one single (N) connection is built on top of several (N − 1) connection, the traffic on the (N) connection being divided between the various (N − 1) connections.

These three types of correspondence between connections in adjacent layers are illustrated in Fig. 9.

### C. Data Transfer

Information is transferred in various types of data units between peer entities and between entities attached to a specific service access point. The data units are defined below and the interrelationship among several of them is illustrated in Fig. 10.

| | Control | Data | Combined |
|---|---|---|---|
| (N) - (N) Peer Entities | (N)-Protocol-Control-Information | (N)-User-Data | (N)-Protocol-Data-Units |
| (N)-(N-1) Adjacent Layers | (N-1)-Interface-Control-Information | (N-1)-Interface Data | (N-1)-Interface Data-Unit |

Fig. 10. Interrelationship between data units.

(N) *Protocol Control Information* is information exchanged between two (N) entities, using an (N − 1) connection, to coordinate their joint operation.

(N) *User Data* is the data transferred between two (N) entities on behalf of the (N + 1) entities for whom the (N) entities are providing services.

An (N) *Protocol Data Unit* is a unit of data which contains (N) Protocol Control Information and possibly (N) User Data.

(N) *Interface Control Information* is information exchanged between an (N + 1) entity and an (N) entity to coordinate their joint operation.

(N) *Interface Data* is information transferred from an (N + 1) entity to an (N) entity for transmission to a correspondent (N + 1) entity over an (N) connection, or conversely, information transferred from an (N) entity to an (N + 1) entity which has been received over an (N) connection from a correspondent (N + 1) entity.

(N) *Interface Data Unit* is the unit of information transferred across the service access point between an (N + 1) entity and an (N) entity in a single interaction. The size of (N) interface data units is not necessarily the same at each end of the connection.

(N − 1) *Service Data Unit* is the amount of (N − 1) interface data whose identity is preserved from one end of an (N − 1) connection to the other. Data may be held within a connection until a complete service data unit is put into the connection.

Expedited (N − 1) service data unit is a small (N − 1) service data unit whose transfer is expedited. The (N − 1) layer ensures that an expedited data unit will not be delivered after any subsequent service data unit or expedited data unit sent on that connection. An expedited (N − 1) service data unit may also be referred to as an (N − 1) expedited data unit.

*Note*: An (N) protocol data unit may be mapped one-to-one onto an (N − 1) service data unit (see Fig. 11).

## V. MANAGEMENT ASPECTS

Even though a number of resources are managed locally, i.e., without involving cooperation between distinct systems, some management functions do.

Examples of such management functions are

configuration information,
cold start/termination,
monitoring,
diagnostics,
reconfiguration, etc.

The OSI Architecture considers management functions as applications of a specific type. Management entities located in the highest layer of the architecture may use the complete set of services offered to all applications in order to perform

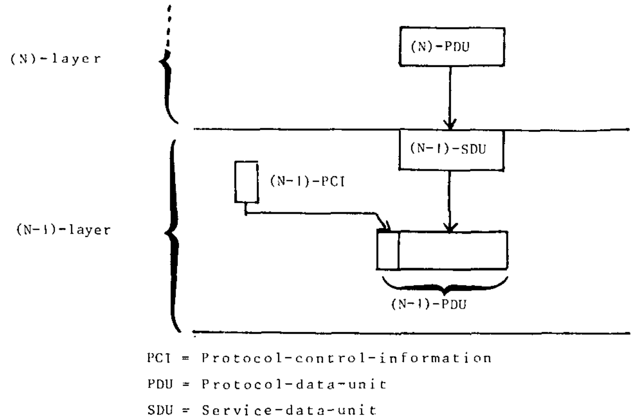



Fig. 11. Logical relationship between data units in adjacent layers.

management functions. This organization of management functions within the OSI Architecture is illustrated in Fig. 12.

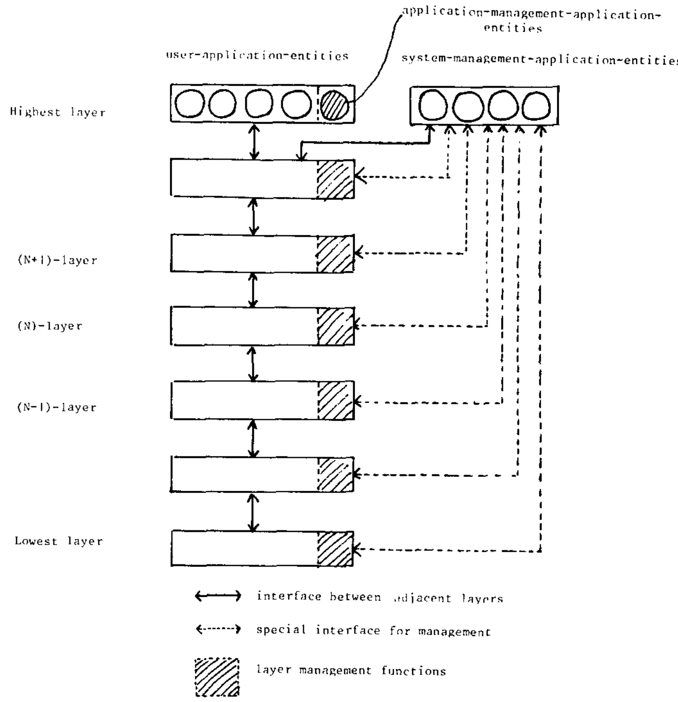



Fig. 12. A representation of management functions.

## VI. THE SEVEN LAYERS OF THE OSI ARCHITECTURE

### *A. Justification of the Seven Layers*

ISO determined a number of principles to be considered for defining the specific set of layers in the OSI architecture, and applied those principles to come up with the seven layers of the OSI Architecture.

Principles to be considered are as follows.

1) Do not create so many layers as to make difficult the system engineering task describing and integrating these layers.

2) Create a boundary at a point where the services description can be small and the number of interactions across the boundary is minimized.

3) Create separate layers to handle functions which are manifestly different in the process performed or the technology involved.

4) Collect similar functions into the same layer.

5) Select boundaries at a point which past experience has demonstrated to be successful.

6) Create a layer of easily localized functions so that the layer could be totaly redesigned and its protocols changed in a major way to take advantages of new advances in architectural, hardware, or software technology without changing the services and interfaces with the adjacent layers.

7) Create a boundary where it may be useful at some point in time to have the corresponding interface standardized.

8) Create a layer when there is a need for a different level of abstraction in the handling of data, e.g., morphology, syntax, semantics.

9) Enable changes of functions or protocols within a layer without affecting the other layers.

10) Create for each layer interfaces with its upper and lower layer only.

11) Create further subgrouping and organization of functions to form sublayers within a layer in cases where distinct communication services need it.

12) Create, where needed, two or more sublayers with a common, and therefore minimum, functionality to allow interface operation with adjacent layers.

13) Allow bypassing of sublayers.

### *B. Specific Layers*

The following is a brief explanation of how the layers were chosen.

1) It is essential that the architecture permits usage of a realistic variety of physical media for interconnection with different control procedures (e.g., V.24, V.35, X.21, etc.). Application of principles 3, 5, and 8 leads to identification of a *Physical Layer* as the lowest layer in the architecture.

2) Some physical communications media (e.g., telephone line) require specific techniques to be used in order to transmit data between systems despite a relatively high error rate (i.e., an error rate not acceptable for the great majority of applications). These specific techniques are used in data-link control procedures which have been studied and standardized for a number of years. It must also be recognized that new physical communications media (e.g., fiber optics) will require different data-link control procedures. Application of principles 3, 5, and 8 leads to identification of a *Data link Layer* on top of the Physical Layer in the architecture.

3) In the Open Systems Architecture, some systems will act as final destination of data. Some systems may act only as intermediate nodes (forwarding data to other systems). Application of principles 3, 5, and 7 leads to identification of a *Network Layer* on top of the Data link Layer. Network-oriented protocols such as routing, for example, will be grouped in this layer. Thus, the Network Layer will provide a connection path (network connection) between a pair of transport entities (see Fig. 13).

4) Control of data transportation from source end system to destination end system (which need not be performed in

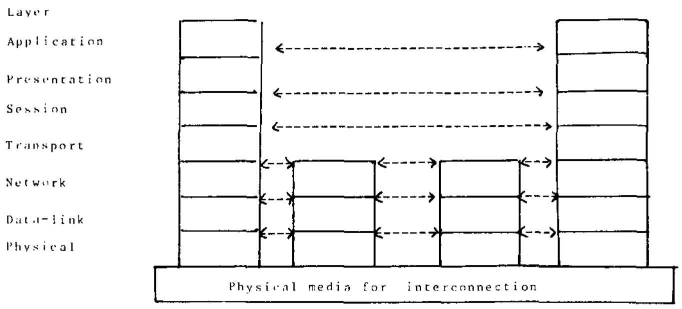



Fig. 13. The seven layers OSI architecture.

intermediate nodes) is the last function to be performed in order to provide the totality of the transport service. Thus, the upper layer in the transport-service part of the architecture is the *Transport Layer,* sitting on top of the Network Layer. This Transport Layer relieves higher layer entities from any concern with the transportation of data between them.

5) In order to bind/unbind distributed activities into a logical relationship that controls the data exchange with respect to synchronization and structure, the need for a dedicated layer has been identified. So the application of principles 3 and 4 leads to the establishment of the *Session Layer* which is on top of the Transport Layer.

6) The remaining set of general interest functions are those related to representation and manipulation of structured data for the benefit of application programs. Application of principles 3 and 4 leads to identification of a *Presentation Layer* on top of the Session Layer.

7) Finally, there are applications consisting of application processes which perform information processing. A portion of these application processes and the protocols by which they communicate comprise the *Application Layer* as the highest layer of the architecture.

The resulting architecture with seven layers, illustrated in Fig. 13 obeys principles 1 and 2.

A more detailed definition of each of the seven layers identified above is given in the following sections, starting from the top with the application layer described in Section VI-C1) down to the physical layer described in Section VI-C7).

### C. Overview of the Seven Layers of the OSI Architecture

1) *The Application Layer*: This is the highest layer in the OSI Architecture. Protocols of this layer directly serve the end user by providing the distributed information service appropriate to an application, to its management, and to system management. Management of Open Systems Interconnection comprises those functions required to initiate, maintain, terminate, and record data concerning the establishment of connections for data transfer among application processes. The other layers exist only to support this layer.

An application is composed of cooperating *application processes* which intercommunicate according to application layer protocols. Application processes are the ultimate source and sink for data exchanged.

A portion of an application process is manifested in the application layer as the execution of application protocol (i.e., application entity). The rest of the application process is considered beyond the scope of the present layered model. Applications or application processes may be of any kind (manual, computerized, industrial, or physical).

2) *The Presentation Layer*: The purpose of the Presentation Layer is to provide the set of services which may be selected by the Application Layer to enable it to interpret the meaning of the data exchanged. These services are for the management of the entry exchange, display, and control of structured data.

The presentation service is location-independent and is considered to be on top of the Session Layer which provides the service of linking a pair of presentation entities.

It is through the use of services provided by the Presentation Layer that applications in an Open Systems Interconnection environment can communicate without unacceptable costs in interface variability, transformations, or application modification.

3) *The Session Layer*: The purpose of the Session Layer is to assist in the support of the interactions between cooperating presentation entities. To do this, the Session Layer provides services which are classified into the following two categories.

a) Binding two presentation entities into a relationship and unbinding them. This is called *session administration service.*

b) Control of data exchange, delimiting, and synchronizing data operations between two presentation entities. This is called *session dialogue service*.

To implement the transfer of data between presentation entities, the Session Layer may employ the services provided by the Transport Layer.

4) *The Transport Layer*: The Transport Layer exists to provide a universal transport service in association with the underlying services provided by lower layers.

The Transport Layer provides transparent transfer of data between session entities. The Transport Layer relieves these session entities from any concern with the detailed way in which reliable and cost-effective transfer of data is achieved.

The Transport Layer is required to optimize the use of available communications services to provide the performance required for each connection between session entities at a minimum cost.

5) *The Network Layer*: The Network Layer provides functional and procedural means to exchange network service data units between two transport entities over a network connection. It provides transport entities with independence from routing and switching considerations.

6) *The Data Link Layer*: The purpose of the Data link Layer is to provide the functional and procedural means to establish, maintain, and release data links between network entities.

7) *The Physical Layer*: The Physical Layer provides mechanical, electrical, functional, and procedural characteristics to establish, maintain, and release physical connections (e.g., data circuits) between data link entities.

## VII. OSI PROTOCOLS DEVELOPMENTS

The model of OSI Architecture defines the services provided by each layer to the next higher layer, and offers con-

*cepts to be used* to specify how each layer performs its specific functions.

Detailed functioning of each layer is defined by the protocols specific to the layer in the framework of the Architecture model.

Most of the initial effort within ISO has been placed on the model of OSI. The next step consists of the definition of standard protocols for each layer.

This section contains a brief description of a likely initial set of protocols, corresponding to specific standardization projects recommended by SC16.

### A. Protocols in the Physical Layer

Standards already exist within CCITT defining:

1) interfaces with physical media for OSI, and

2) protocols for establishing, controlling, and releasing switched data circuits.

Such standards are described in other papers in this issue [9], [10], e.g., X.21, V.24, V.35, etc.

The only work to be done will consist of clearly relating those standards to the OSI Architecture model.

### B. Protocols in the Data Link Layer

Standard protocols for the Data link Layer have already been developed within ISO, which are described in other papers within this issue [11], [12].

The most popular Data link Layer protocol is likely to be HDLC [13], without ruling out the possibility of using also other character-oriented standards.

Just as for the Physical Layer, the remaining work will consist mainly of clearly relating these existing standards to the OSI Architecture model.

### C. Protocols in the Network Layer

An important basis for protocols in the network layer is level 3 of the X.25 interface [14] defined by CCITT and described in another paper in this issue. It will have to be enhanced in particular to permit interconnection of private and public networks.

Other types of protocols are likely to be standardized later in this layer, and in particular, protocols corresponding to Datagram networks [10].

### D. Protocols in the Transport Layer

No standard exists at present for this layer; a large amount of experience has been accumulated in this area and several proposals are available.

The most widely known proposal is the Transport Protocol proposed by IFIP and known as INWG 96.1 [15], which could serve as a basis for defining an international standard.

### E. Protocols for the Session Layer

No standard exists and no proposal has been currently available, since in most networks, session functions were often considered as part of higher layer functions such as Virtual Terminal and File Transfer.

A standard Session Layer Protocol can easily be extracted from existing higher layer protocols.

### F. Presentation Layer Protocol

So far, Virtual Terminal Protocols and part of Virtual File are considered the most urgent protocols to be developed in the Presentation Layer.

A number of VTP's are available (e.g., [16], [17]), many of them being very similar, and it should be easy to derive a Standard VTP from these proposals, also making use of the ISO standard for "Extended Control Characters for I/O Imaging Devices" [18]. These protocols are reviewed in another paper in this issue [19].

The situation is similar for File Transfer Protocols.

### G. Management Protocols

Most of the work within ISO has been done so far on the architecture of management functions, and very little work has been done on management protocols themselves. Therefore, it is too early to give indications on the likely results of the ISO work in this area.

## VIII. CONCLUSION

The development of OSI Standards is a very big challenge, the result of which will impact all future computer communication developments. If standards come too late or are inadequate, interconnection of heterogeneous systems will not be possible or will be very costly.

The work collectively achieved so far by SC16 members is very promising, and additional efforts should be expended to capitalize on these initial results and come up rapidly with the most urgently needed set of standards which will support initial usage of OSI (mainly terminals accessing services and file transfers). The next set of standards, including OSI management and access to distributed data, will have to follow very soon.

Common standards between ISO and CCITT are also essential to the success of standardization, since new services announced by PTT's and common carriers are very similar to data processing services offered as computer manufacturer products, and duplication of now compatible standards could simply cause the standardization effort to fail. In this regard, acceptance of the OSI Reference Model by CCITT Rapporteur's Group on Layered Architecture for Public Data Networks Services is most promising.

It is essential that all partners in this standardization process expend their best effort so it will be successful, and the benefits can be shared by all users, manufacturers of terminals and computers, and the PTT's/common carriers.

## ACKNOWLEDGMENT

The OSI Architecture model briefly described in this paper results from the work of more than 100 experts from many countries and international organizations. Participation in this collective work was really a fascinating experience for the author who acknowledges the numerous contributions from SC16 members which have been merged in the final version of the OSI Architecture briefly presented here.

## REFERENCES


[1] L. G. Roberts and B. D. Wessler, "Computer network development to achieve resource sharing," in *Proc. SJCC*, 1970, pp. 543–549.

[2] L. Pouzin, "Presentation and major design aspects of the CYCLADES computer network," in *Proc. 3rd ACM-IEEE Commun. Symp.*, Tampa, FL, Nov. 1973, pp. 80–87.

[3] J. H. McFayden, "Systems network architecture: An overview," *IBM Syst. J.*, vol. 15, no. 1, pp. 4–23, 1976.

[4] G. E. Conant and S. Wecker, "DNA, An Architecture for heterogeneous computer networks," in *Proc., ICCC*, Toronto, Ont., Canada, Aı 1976, pp. 618–625.

[5] H. Zimmermann, "High level protocols standardization: Technical and political issues," in *Proc. ICCC*, Toronto, Ont., Canada, Aug. 1976, pp. 373–376.

[6] ISO/TC97/SC16, "Provisional model of open systems architecture," Doc. N34, Mar. 1978.

[7] ISO/TC97/SC16, "Reference model of open systems interconnection," Doc. N227, June 1979.

[8] H. Zimmermann and N. Naffah, "On open systems architecture," in *Proc. ICCC*, Kyoto, Japan, Sept. 1978, pp. 669–674.

[9] H. V. Bertine, "Physical level protocols," this issue pp. 433–444.

[10] H. C. Folts, "Procedures for circuit-switched service in synchronous public data networks," and "X.25 transaction-oriented features—Datagram and fast select," this issue, pp. 489–496.

[11] J. W. Conard, "Character oriented data link control protocols," this issue, pp. 445–454.

[12] D. E. Carlson, "Bit-oriented data link control procedures," this issue, pp. 455–467.

[13] ISO, "High level data link control-elements of procedure," IS 4335, 1977.

[14] CCITT, "X25," *Orange Book*, vol. VIII-2, 1977, pp. 70–108.

[15] IFIP-WG 6.1, "Proposal for an internetwork end-to-end transport protocol," INWG Note 96.1; also, doc. ISO/TC97 SC16/N24, 46 pp., Mar. 1978.

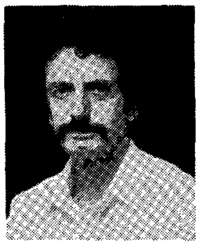
[16] IFIP-WG 6.1, "Proposal for a standard virtual terminal protocol," doc. ISO/TC97/SC16/N23, 56 pp., Feb. 1978.

[17] EURONET, "Data entry virtual terminal protocol for EURONET," VTP/D-Issue 4, doc. EEC/WGS/165.

[18] ISO, "Extended control characters for I/0 imaging devices," DP 6429.

[19] J. Day, "Terminal protocols," this issue, pp. 585–593.


★




**Hubert Zimmermann** received degrees in engineering from Ecole Polytechnique, Paris, France, in 1963, and from Ecole Nationale Superieure des Telecommunications, Paris, France, in 1966.

He is presently in charge of the computer communications group at IRIA, Rocquencourt, France. He was involved in development of command and control systems before joining IRIA in 1972 to start the CYCLADES project with L. Pouzin. Within CYCLADES, he was mainly responsible for design and implementation of host protocols.

Dr. Zimmermann is a member of IFIP WG 6.1 [International Network Working Group (INWG)]. He also chaired the Protocol Subgroup and co-authored several proposals for international protocols. He is an active participant in the development of standards for Open Systems Interconnection (OSI) within ISO, where he chairs the working group on OSI architecture.