INTERNATIONAL TELECOMMUNICATION UNION

ITU-T Q.1400

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(03/93)

INTELLIGENT NETWORK

ARCHITECTURE FRAMEWORK FOR THE DEVELOPMENT OF SIGNALLING AND OA&M PROTOCOLS USING OSI CONCEPTS

ITU-T Recommendation Q.1400

(Previously "CCITT Recommendation")

FOREWORD

The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the International Telecommunication Union. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, established the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

ITU-T Recommendation Q.1400 was prepared by the ITU-T Study Group XI (1988-1993) and was approved by the WTSC (Helsinki, March 1-12, 1993).

NOTES

1 As a consequence of a reform process within the International Telecommunication Union (ITU), the CCITT ceased to exist as of 28 February 1993. In its place, the ITU Telecommunication Standardization Sector (ITU-T) was created as of 1 March 1993. Similarly, in this reform process, the CCIR and the IFRB have been replaced by the Radiocommunication Sector.

In order not to delay publication of this Recommendation, no change has been made in the text to references containing the acronyms "CCITT, CCIR or IFRB" or their associated entities such as Plenary Assembly, Secretariat, etc. Future editions of this Recommendation will contain the proper terminology related to the new ITU structure.

2 In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

CONTENTS

SUMMARY

This Recommendation provides information on key concepts of the Open Systems Interconnection (OSI) Reference Model and how these concepts are applied in various portions of Signalling System No. 7 (SS No.7) and Digital Subscriber Signalling System No. 1 (DSS 1). These concepts form the basis for the development of new application protocols within the SS No. 7 and DSS 1 environments. They apply equally to Operations, Administration and Management protocols.

The discussion covers the Application Layer Structure (ALS), the nature of the services provided by the Association Control Service Element in the Application Layer and how it may be adapted to the signalling environment, and the services provided by the OSI Presentation Layer.

The application of the concepts is illustrated by means of a detailed discussion of their use in the development of the Intelligent Network Application Part (INAP) for Capability Set 1, Recommendation Q.1218.

In addition, this Recommendation contains guidelines for use when an existing protocol is extended. These are presented as two sets. One is for existing, non-OSI structured protocols, and the other is for Remote Operations Service Element (ROSE)-based protocols. ROSE-based protocols are presently the most widely used OSI protocols within telephony signalling systems.

ARCHITECTURE FRAMEWORK FOR THE DEVELOPMENT OF SIGNALLING AND OA&M PROTOCOLS USING OSI CONCEPTS

(Helsinki, 1993)

1 General

1.1 Purpose

This Recommendation provides a framework for the common development and evolution of protocol specifications using OSI concepts, and to provide guidance on techniques that should be applied to the detailed specification of signalling and OA&M protocols.

1.2 Scope

The framework and guidance contained in this Recommendation apply to all signalling protocols, including those used to access network resources as well as those used within a network to provide services to users of the network.

This Recommendation is applicable to emerging signalling protocols, providing the framework and guidance for their specification. Examples include the Intelligent Network Application Part (INAP), the B-ISDN application signalling protocol and the Telecommunications Management Network (TMN) protocols.

This Recommendation is also applicable to the evolution of existing message-based signalling protocols, such as Digital Subscriber Signalling System No. 1 (DSS 1), Transaction Capabilities (TC), Operations Administration and Maintenance and Administration Part (OMAP), Integrated Services User Part (ISUP), Telephony User Part (TUP), Signalling Connection Control Part (SCCP), and Message Transfer Part (MTP).

This Recommendation is not intended to take precedence over other specifications which describe the details of specific topics discussed herein. Where discrepancies or inconsistencies occur, the referenced specification should be taken as definitive. It is intended that where such discrepancies are uncovered, they will be addressed jointly with experts in the area affected with the intent to reach consensus such that the discrepancy or inconsistency is removed in future versions of this Recommendation.

1.3 Background

As of the 1988 set of Recommendations, the work on signalling protocols had not proceeded within a common framework and set of guidelines. This has resulted in the development of individual protocol architectures which are not well aligned. In addition, different environments for the application of a protocol have led to decisions specific to the environment which have, from time to time, led to interworking difficulties when transitioning from one environment to another. While, in general, these difficulties have been overcome, they have highlighted the need for a common protocol architecture framework together with guidelines for its application.

In the early stages of the work that led to the existing (1988 set of Recommendations) message-based signalling protocols, work on OSI concepts, most particularly the seven layer communications model, was incomplete. This resulted in some parallel protocol modelling work which has not been well integrated.

Since the work on message-based signalling protocols got under way (SS No. 6; 1980 Recommendations as first specification of SS No. 7), physical technology advances have contributed major enhancement in:

- processing power (instructions executed per unit time);
- memory capacity;

– physical media capacity (bit rate); and

– performance of physical media (bit error rate, down-time).

Software technology advances have also occurred:

– maturity of OSI model;

– specification of layer services and protocols;

– structured programming techniques;

– higher level languages; and

– distributed processing techniques.

The specification of many of the existing message-based signalling protocols is considered flawed because they do not clearly distinguish the application process specification from the protocol specification. That is, the existing specifications are a combination of application procedures and supporting protocols without a clear distinction between the two. This situation leads to significant difficulties in extending or evolving the protocols when new application procedures are required. Note that this area advanced substantially from SS No. 6 to SS No. 7 through the distinction achieved between the SS No. 7 MTP and users of the MTP. The recognition of the appropriateness of distinguishing the application process specification from the application protocol specification is also reflected in the present work on the Integrated Services Control Part (ISCP).

As realization of the problems with the existing (1988 set of Recommendations) message-based signalling protocols has emerged, there has also been a realization that the parallel work on OSI has matured and that it forms a basis for communications protocols in general.

1.4 OSI applicability

Despite their inception at approximately the same time, OSI and ISDN have not significantly influenced each other's models. Two different principles drove the development of OSI and ISDN protocols, mainly because of the perceived differences between the data communications and the telecommunications environments. In particular, the main requirements of the telecommunications signalling environment has been efficiency, while the data processing environment's main emphasis has been "openness". "Openness" is the ability for any user with the communications capabilities provided by the OSI-standardized protocols to access the widest variety of applications subject to administrative restrictions.

OSI provides a reference model, which is a framework or discipline for providing a communications infrastructure that may be used by any application in a distributed environment. It also provides a set of common protocol standards which provide uniform communications capabilities independent of the precise nature of the application.

There is a significant advantage to be obtained by studying the OSI models and protocols. The evolution of telephone networks requires ever more exchange of information among software controlled devices (computers). The telecommunication industry is solving similar problems and should take advantage of the knowledge and large investment represented by OSI.

1.5 Relationship to the Three Stage Process

This subclause includes an outline of the three stage process defined in Recommendations I.130 and Q.65. The three stage process was designed for the complete definition and specification of individual ISDN (and non-ISDN) services. It provides, as described below, a stage for the specification of service specific protocol. It is anticipated that further evolution of telecommunications networks will include significant adoption of Intelligent Network (IN) techniques and capabilities. IN represents a generalization of the service specific work being done on a number of supplementary services with the aim of achieving standards. The generalization of service work will also require generalization of the protocol. A major objective of the protocol architecture guidance is to ensure a well-ordered, open-ended structure and framework for these general protocols. This will enable the protocols built on this framework to evolve and be extended in a straightforward manner with minimal version and interworking problems.

The three stage process may be summarized as:

- Stage 1 is an overall service description from the user's standpoint.
- Stage 2 is an overall description of the organization of the network functions to map service requirements into network capabilities.
- Stage 3 is the definition of switching and signalling capabilities needed to support services defined in Stage 1.

Each stage consists of several steps.

Stage 1

Stage 1 is an overall service description from the user's point of view, but does not deal with the details of the human interface itself. The Stage 1 service description is independent of the amount of functionality in the user's terminal, other than that required to provide the human interface. For example the conference calling service description is designed to be independent of whether the conference bridge is in the terminal, in the serving exchange or elsewhere.

The steps in Stage 1 are:

- *Step 1.1* – Service prose definition and description.
- *Step 1.2* – Static description of the service using attributes.
- *Step 1.3* – Dynamic description of the service using graphic means.

Stage 2

Stage 2 identifies the functional capabilities and the information flows needed to support the service as described in Stage 1. The Stage 2 description will also include user operations not directly associated with a call (e.g. user change of call forwarding parameters via his service interface) as described in Stage 1. Furthermore, it identifies various possible physical locations for the functional capabilities. The output of Stage 2 which is signalling system independent is used as an input to the design of signalling system and exchange switching Recommendations.

The steps in Stage 2 are:

- *Step 2.1* – Derivation of a functional model.
- *Step 2.2* – Information flow diagrams.
- *Step 2.3* – SDL diagrams for functional entities.
- *Step 2.4* – Functional entity actions.
- *Step 2.5* – Allocation of functional entities to physical locations.

Stage 3

In Stage 3 the information flow and SDL diagrams from the Stage 2 output form the basis for producing the signalling system protocol Recommendations and the switching Recommendations.

Stage 3 will need to be repeated for each service where, because of different allocations of functional entities to physical locations, different protocols and procedures are needed.

The protocol architecture guidelines included in this Recommendation have been prepared based on known and predicted relationship requirements.

It is expected that these protocol architecture guidelines will evolve to include further structure and capability as relationships are identified and specified that require more complex capabilities than initially provided.

2 The OSI Reference Model

2.1 General Description of the OSI Reference Model

This subclause provides some general remarks on the OSI model. Later subclauses address the Application Layer of that model, together with related aspects, in some detail.

The purpose of the Reference Model of Open Systems for CCITT Applications (Recommendation X.200) is to provide a well-defined structure for modelling the interconnection and exchange of information between users in a communication system. The approach allows standardized procedures to be defined not only to provide an open system interconnection between users over a single network, but also to permit interworking between networks to allow communication between users over several networks in tandem.

The approach taken in the OSI Reference Model is to partition the model used to describe the interconnection and exchange of information between users in a communication system into seven layers. From the point of view of a particular layer, the lower layers provide a "transfer" service with specific features. The way in which the lower layers are realized is immaterial to the next higher layers. Correspondingly, the lower layers are not concerned with the meaning of information coming from higher layers or the reasons for its transfer.

The characteristics of each layer are described below:

a) *Physical Layer (Layer 1)*[1] – Provides transparent transmission of a bit stream over a circuit built in some physical communication medium. It furnishes the interface to the physical media and is responsible for relaying bits (i.e. interconnects data circuits). A 64 kbit/s link as used for SS No. 7 is an example.

b) *Data Link Layer (Layer 2)* – Overcomes the limitations inherent in the physical circuits and allows errors in transmission to be detected and recovered, thereby masking deficiencies in transmission quality.

c) *Network Layer (Layer 3)* – Transfers data transparently by performing routing and relaying of data between end users. One or more of the subnetworks may interwork at the Network Layer to provide an end user to end user network service. A connectionless network provides for the transfer of data between end users, making no attempt to guarantee a relationship between two or more messages from the same user.

d) *Transport Layer (Layer 4)* – Provides an end user to end user transfer optimizing the use of resources (i.e. network service) according to the type and character of the communication, and relieves the user of any concern for the details of the transfer. The Transport Layer always operates end-to-end, enhancing the Network Layer when necessary to meet the Quality of Service objectives of the users.

e) *Session Layer (Layer 5)* – Co-ordinates the interaction within each association between communicating application processes. Full and half duplex dialogues are examples of possible Session Layer modes.

f) *Presentation Layer (Layer 6)* – Transforms the syntax of the data which is to be transferred into a form recognizable by the communicating application processes.

g) *Application Layer (Layer 7)* – Specifies the nature of the communication required to satisfy the users' needs. This is the highest layer in the Model and so does not have a boundary with a higher layer. The Application Layer provides the sole means for application processes to access the OSI environment.

2.2 OSI Layering and SS No. 7

Evolution of SS No. 7 architecture has been based on the Open Systems Interconnection (OSI) Reference Model (see 2.1). OSI considers primarily connection-oriented protocols, that is, protocols which establish a logical connection before transferring data. The Network Service Part (NSP) of SS No. 7 provides both connectionless and connection-oriented protocol. The NSP of SS No. 7 evolved from a four-level model, with the lower three levels corresponding to the lower three layers of the OSI Reference Model, and level 4 corresponding to users of the lower three levels but without further generalized internal structure.

Layers 1-3 comprise functions for the transportation of information from one location to another, possibly via a number of communication links in tandem. These functions provide the basis on which a communication network can be built.

---

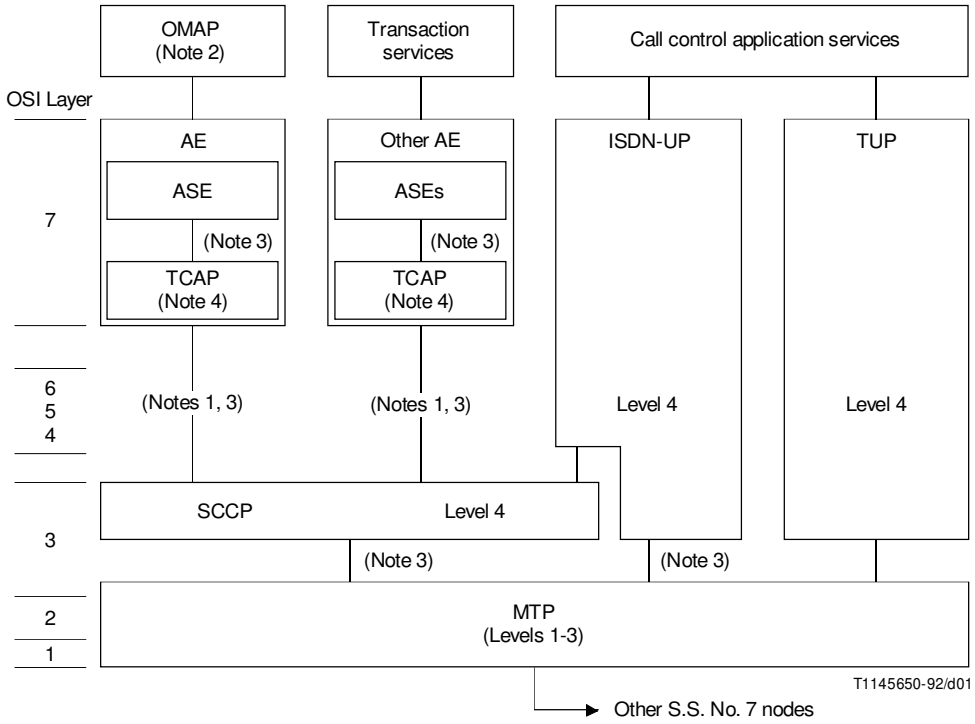
[1] While OSI does not refer to its layers by numbers, it has become common usage to number the layers. This Recommendation uses the name of the layer or its number interchangeably.

The SCCP provides, with the MTP, OSI Layers 1-3.

Layers 4-7 define functions relating to end-to-end communication. These layers are so defined that they are independent of the internal structure of the communication network.

Transaction Capabilities directly uses the Network Service provided by the connectionless SCCP. The ISCP allows for the possibility of functions in Layers 4-6, particularly at Layer 6. Other SS No. 7 application protocols, e.g. ISUP and TUP, do not provide for such an explicit structure.

Figure 1 illustrates the architecture of SS No. 7.



OMAP	Operations, Maintenance and Administration Part
AE	Application Entity
ASE	Application Service Element
TCAP	Transaction Capabilities Application Part
ISDN-UP	ISDN User Part
TUP	Telephony User Part
SCCP	Signalling Connection Control Part
MTP	Message Transfer Part

NOTES

1 The only standardized user of this interface is TCAP using the services of the connectionless SCCP.

2 OMAP is SS No. 7 management.

3 SS No. 7 primitive interface.

4 TCAP may be considered as an ASE.

FIGURE 1/Q.1400

Relationship Between SS No. 7 Functional Levels and OSI Layering

3 Control and User Plane Modelling Aspects

This clause supplements the material contained in Recommendation I.324.

As discussed in Recommendation I.324, the interaction between a terminal and an exchange may be modelled using OSI concepts. The terminal and exchange in general interact with each other on a peer-to-peer basis. The interaction is in the control plane and concerns the provision of a resource in the user plane (e.g. the Physical Layer channel in the case of a voice circuit). For example, in circuit mode, this resource is at Layer 1 (once established by the network) and the user at each end must provide Layers 2 through 7 (described in clause 2) according to his needs. The Layer 1 bearer channel provided by the network needs to be understood as entirely distinct (in the logical sense) from the Layer 1 being used to transport control plane messages. Further, the term "network" as generally used in telephony does not have the same connotation as the term in OSI. The telephone network is a physical network made up of exchanges and interconnecting bearer channels, and is the equivalent of the OSI term "subnetwork". The OSI term "Network" refers to the Layer 3 entity which has responsibilities including routing and relaying of messages on behalf of users of the network towards indicated destinations.

The DSS 1 term "Layer 3" should not be confused with the OSI Network Layer (sometimes referred to as Layer 3). DSS 1 Layer 3 has aspects of OSI Layers 3 to 7 in the control plane. It is therefore incorrect to place the terminal at the Network Layer as is sometimes done. Rather it should be viewed in two ways. For control plane purposes, it is a full Application Process with an Application Entity for its communication needs (further details on these concepts may be found in clause 4). In the user plane, the terminal provides the Application Entity but not the remainder of the Application Process. The remainder of the Application Process is provided by the human user of the terminal and interfaces to it via the man-machine interface (MMI). Alternatively, the user may be a computer interacting with the terminal via a machine-to-machine interface. After physical path establishment, the computer may itself provide Layer 2 through 7 functions.

The discussion in this Recommendation refers to structure and addressing aspects of signalling protocols in the control plane. The user plane has its own addressing mechanisms (e.g. Recommendation E.164 address or sub-address).

Further discussion and modelling in this area may be found in Recommendation I.324.

4 OSI Application Layer Structure[2)]

The following is a review of key concepts of the OSI Application Layer Structure described in ISO/IEC 9545.

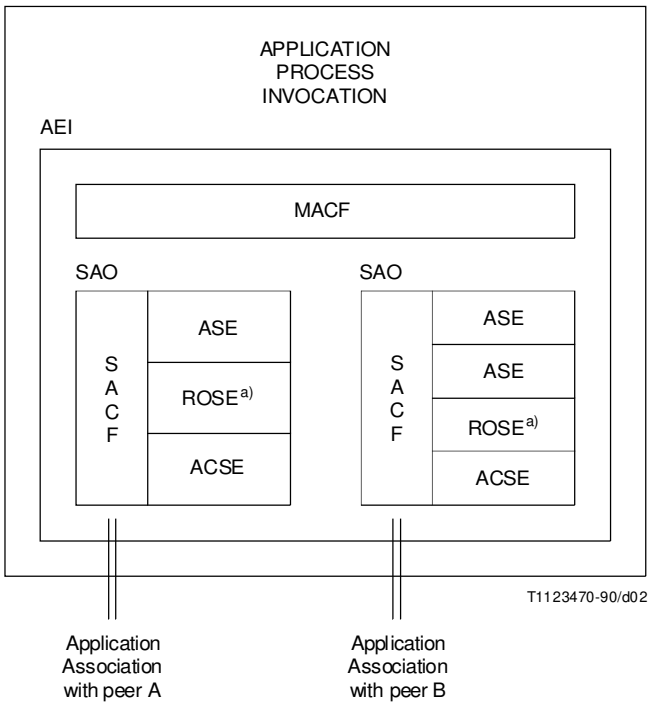
The structure of the OSI Application Layer, Layer 7, is different from that of any other layer in the OSI Reference Model. Whereas each of the other six layers contains a set of well defined functions within a monolithic layer structure, the OSI Application Layer is structured modularly to allow flexibility in function and form, to meet the communication needs of every possible distributed application. This difference arises from the role of the Application Layer as the bridge between the work of Application Processes (of which it is a part) and the work of the OSI lower layers.

The Application Layer must be able to perform the functions necessary to communicate any information the Application Process needs conveyed to a remote peer. Thus, unlike the other layers in the OSI model, the Application Layer must provide functions that are application specific. The form and content of the functions in the Application Layer are dependent on the needs of the Application Process using these functions. In contrast, lower layers in the OSI stack provide a fixed set of functions, which may be manipulated as needed, but not changed or expanded upon. In order to provide flexibility and ease of expansion, the Application Layer has to be defined in an open-ended way, with room for Application specific functions, yet still enforce standard methods of communication.

To accomplish all this, the structure shown in Figure 2 was conceived for the OSI Application Layer. The abbreviations in the figure are expanded in subsequent subclauses.

---

2) The ongoing work on Extended Application Layer Structure needs to be considered in the discussion of OSI applied to signalling systems.



a) Example of an ASE commonly used.

AEI	Application Entity Invocation
MACF	Multiple Association Control Function
SAO	Single Association Object
SACF	Single Association Control Function
ASE	Application Service Element
ROSE	Remote Operations Service Element
ACSE	Association Control Service Element

FIGURE 2/Q.1400

The OSI Application Layer Structure

This is a very modular approach, with each function in the OSI layer neatly labelled and boxed. Thus, it is easy to include the appropriate functions, such as a highly application specific function (such as an account management function), while keeping within a structured framework.

4.1 AEs, APs, AEIs and APIs

An Application Entity or AE is the function that an Application Process (AP) uses to communicate with its peers. An AP can use several AEs, each of which provides a specific set of communication functions for the AP. An AE is composed of definitions of each of the functions and the rules that govern the use of these functions.

The AE and AP are abstract entities whose functions may be thought of as being realized through software programs. Thus, when instances of each are created and performing functions, the word “invocation” is added to the title. An actual instance of an AE is an AE-Invocation or AEI and the instance of an AP is an AP-Invocation or API.

An API may have many AEIs performing communication functions for it, but the coordination of such AEIs is up to the API itself.

4.2 AE-type and Application Context

The AE/AEI relationship may be viewed as a type/instance relationship. A type is a definition of a class of objects. Examples of types are "integer", "elm tree" or "automobile". Instances of types are particular objects within the class, such as "42", "the elm tree in town square", or "my car". Thus, one can view an AE as a definition of a type, with an instance of that type being the AEI.

An AEI is the abstraction of an actual "run-time" program that performs all or a subset of the communication functions defined by the AE-type specifications. The actual procedures that will be performed or need to be performed for an instance of communication are determined by the Application Context. While an AE-type defines a set of functions used for communication, an actual instance of communication may require that only a subset of these functions be performed. The Application Context is used just to state which functions are needed, and based on this information, the AEI that fits these criteria is instantiated. Different Application Contexts may be handled by instances of the same AE-type, as long as the AE-type encompasses all the functions needed by all the requested Application Contexts.

4.3 ASEs, SACFs and MACFs

The basic component of the AE is an ASE or Application Service Element. An ASE is an element that defines a function or set of functions to help accomplish application communication. The number or set of functions in an ASE is determined by the designer of the application protocol. Thus, one might think of the AE as a large computer program, made up of many sub-procedures (ASEs). How the program is split up into sub-procedures is purely up to the implementor, based on ease of programming and debugging.

How the communication functions are divided among ASEs is the responsibility of ASE designers (i.e. Application Layer Standards groups).

In OSI, several ASEs have been standardized, so that a designer may pick and choose the set of ASEs needed for a particular Application Process communication. There are ASEs for File Transfer and Access Management (FTAM), Message Handling Systems (MHS), Common Management Information Protocol (CMIP), Transaction Processing (TP), etc. In particular, there are two ASEs that are of special interest. The Association Control Service Element (ACSE) is a special ASE that is always included in the set of ASEs chosen by a designer. This ASE sets up and releases Application Associations, over which AEIs exchange information. An Application Association is a logical relationship between the two peer Application Layer entities (e.g. AEIs). The Application Layer entities exchange protocols over Associations which make use of underlying Presentation Layer connections. There is a one-to-one relationship between Application Associations and Presentation Connections.

The other ASE that is of immediate interest is the Remote Operations Service Element (ROSE). This ASE offers a generic remote procedure call facility. ROSE provides the framework for invoking remote procedures and returning the results of these procedures. ROSE identifies remote procedures using the term OPERATION. ROSE does not itself determine which particular operations may be invoked, but merely provides the framework for requesting and responding to application specific operations. Thus, ROSE is very general-purpose and has been adopted for use in a variety of application protocols (such as CMIP, MHS, TCAP, and Q.932).

Once a set of ASEs has been assembled (to be used in a single communication with a peer), including exactly one ACSE ASE, there may be a need for rules to guide the joint use of these ASEs. For example, the first ASE to be used must be the ACSE ASE, since an application association must be set up before any other communication can be achieved. Thus, a rule may be that no other ASE may be used other than ACSE until an Application Association has been brought up[3]. These types of rules are contained in a Single Association Control Function (SACF). The SACF represents the rules and regulations governing the use of the ASEs that are being used for communication over a single Application Association to a peer.

---

3) This area is further discussed in relation to the application of these concepts to signalling needs.

There may also be rules that govern multiple communications with many peers. Thus, an Application Process Invocation may have a need to communicate with more than one peer over more than one Application Association. For example, suppose that an Application Process Invocation is communicating with one peer in order to debit a bank account $50, while communicating with another peer to credit another bank account for that same $50. The API would not wish to debit the first account until it was sure that the amount was credited to the other account. Thus a rule for coordinating the two communications would be that if either one of the tasks were to fail, the other task would also be forced to fail. The entire transaction would then fail. (This is an example taken from TP – Transaction Processing.) The Multiple Association Control Function or MACF represents the rules and regulations governing the coordination of the set of peer-to-peer communications within an AEI.

The combination of the ASEs, the SACF rules and the MACF rules form the total definition of the AE-type. An Application Context is then used to establish which functions are to be used for a particular instance of communication. These functions are performed by an AEI over a single Application Association. This is further described in clause 6.

4.4 SAOs

The collection of the particular set of ASEs and SACF rules to be used over one association is called a Single Association Object (SAO). An SAO is the representation of the functions that are needed to communicate over a single Application Association to a peer. An AEI may contain many SAOs, all based on the same AE-type, but each possibly performing different sets of functions based on different Application Contexts. At a minimum, an AEI may contain no SAOs or one SAO and may or may not contain a MACF. At the other extreme, an AEI might contain a very large number of SAOs, offering different subsets of the functions defined in the AE-type, with an MACF governing the interactions among the SAOs.

ASEs and Application Contexts are standardized so that Application Processes may make use of them.

5 Addressing

5.1 Introduction

The Transaction Capabilities (TC) portion of the SS No. 7 Application Layer evolved towards an architecture which used concepts such as Application Entity (AE) and Application Service Element (ASE) that are defined in the OSI Application Layer Structure (ALS) standard, ISO/IEC 9545. A similar formalization has occurred in the ISCP for the protocol stack associated with call/bearer-related signalling.

However, a complete alignment with the OSI ALS is currently not possible because the SS No. 7 protocol architecture does not support a key OSI requirement which is the concept of an explicit association between peer AE-Invocations supported by an underlying Presentation Layer connection. This is due to the absence from SS No. 7 of the Intermediate Service Part (ISP) which is the collection of services provided by the OSI Transport, Session and Presentation Layers. Another issue, also arising because of the absence of the ISP, which requires considerable clarification, is the question of how SS No. 7 applications are addressed. An OSI Application Process would be accessed through an AE which is addressed by a Presentation Service Access Point (PSAP). The absence of an explicit Presentation Layer in SS No. 7 suggests that whatever addressing information is currently available in SS No. 7 indirectly provides a Presentation Address.

This subclause clarifies aspects of the SS No. 7 protocol architecture. Its aim is to examine addressing concepts and functions in SS No. 7 and OSI. This will allow a common basis for comparison particularly when discussing questions of alignment of the two protocol architectures. Such considerations are particularly applicable to the work on the “ISDN Signalling Control Part”.

The next two subclauses explore the relationships between existing SS No. 7 addressing information and those defined in the OSI Naming and Addressing specifications, Recommendation X.650 as well as Recommendation X.213 on Network Layer addressing.

Addressing equivalents for DSS 1 are also examined.

5.2 Basic Definitions of SS No. 7 Addressing Information

The various addressing information elements present in an SS No. 7 message, using the definitions given in Recommendation Q.700 are:

a) **Point Code (PC)** – This uniquely identifies a node in an SS No. 7 network. It is used for inter-nodal, intra-network addressing in conjunction with the 2-bit Network Indicator field of the Service Information Octet defined below.

b) **Service Information Octet (SIO)** – This consists of a 4-bit Service Indicator (SI) and 2-bit Network Indicator. The SI is being used by a signalling point's distribution function to determine the "user" of the incoming message. The Service Indicator addresses "users" of the Message Transfer Part. Examples of "users" are the Signalling Connection Control Point (SCCP), ISDN User Part (ISDN-UP) and the Telephone User Part (TUP).

c) **Global Title (GT)** – This is addressing used by the SCCP, comprising dialled digits or another form of address that will not be recognized by the SS No. 7 Network Layer. Therefore, translation of this information to an SS No. 7 Network Address is necessary.

d) **Sub-System Number (SSN)** – This identifies a sub-system accessed via the SCCP within a node and may be a User Part (e.g. ISDN-UP, SCCP Management) or an Application Entity containing the TCAP ASE.

5.3 Addressing Information in DSS 1

DSS 1 is a protocol for use between an exchange and an ISDN terminal or between PABXs. It therefore is not "networked" in the OSI sense. In fact, it may be modelled either as lacking a Network Layer altogether, or alternatively as representing a very small closed network.

DSS 1 supports identification of a terminal on an ISDN interface through a TEI or Terminal Endpoint Identifier. It further provides for distinguishing among classes of procedures through the use of SAPIs or Service Access Point Identifiers. SAPIs indicate, for example, B-channel bearer control signalling or D-channel packet data.

5.4 A Brief Review of OSI Addressing Concepts

The relevant definitions from Recommendation X.650 which will help determine the necessary mapping of concepts and terminology are provided next.

In general, an (N)-Address is defined as a set of (N)-Service Access Points [(N)-SAPs] where (N) refers to any OSI layer and an SAP is the conceptual Interface point through which a Layer (N+1) entity issues/receives service primitives to/from a Layer (N) entity during an instance of communication. An (N)-SAP Address is used in the case when the (N)-Address identifies only one SAP. Therefore, (N)-Addresses are used to identify sets of (N)-SAPs in order to locate (N+1)-entities.

Each (N)-SAP in the set identified by an (N)-Address is bound to (N+1)-entities of the same type, i.e. each of these (N+1)-entities provides the same functions. Within an (N)-Layer an (N)-selector is used to identify a (set of) (N)-SAP(s), i.e. to address an (N+1)-entity once the end open system has been unambiguously identified. A locally chosen (N)-selector value would be known to communicating open systems either through directory look-up or advertisement and exchanged as part of the (N)-Protocol Addressing Information [(N)-PAI] during connection establishment.

When an actual connection is established between two peer (N+1)-entity Invocations, each assigns a local (N)-connection-endpoint-identifier [(N)-CEI] to that particular instance of communication. Thereafter, the (N)-CEI is sufficient addressing information during the data transfer phase.

Specifically, at the Network Layer, a Network Address is in general a set of NSAPs (where each NSAP is structured as a part that identifies the Network Entity unambiguously in the open system environment) plus a locally-specified "selector" which chooses a particular Network Service Access Point (NSAP). Transport entities of the same type are bound to each NSAP within this Network Address and, in general, different Transport entities are bound to different Network Addresses. The most common sort of Network Address is an NSAP Address which is a Network Address consisting of only one NSAP. At connection establishment time, individual network connections between instances of peer transport entities are assigned local connection endpoint identifiers, which are then used as addressing information during the subsequent data transfer phase.

Recommendation X.213 has defined the structure and abstract syntax of an NSAP address leaving the actual encodings to specific Network Layer protocols standards. This structure is shown in Figure 3.

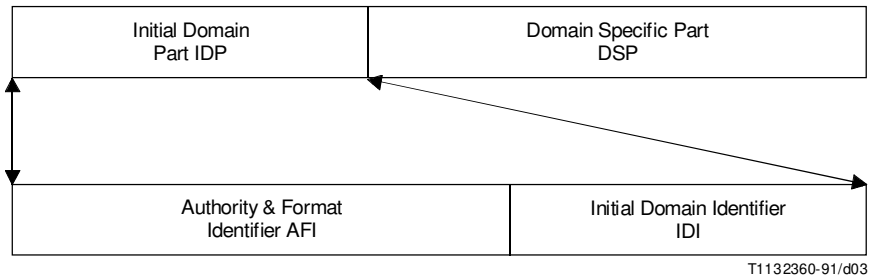


FIGURE 3/Q.1400

OSI NSAP Address Structure

The conceptual format of the NSAP address is hierarchical in that the initial part of the address, the IDP, unambiguously identifies an addressing domain, while the rest, the DSP, is allocated by the authority identified by that addressing domain. The IDP is further structured into two parts: the first part, the AFI, names the addressing authority (e.g. ISO or CCITT) responsible for allocating values of the second part, the IDI, as well as the abstract syntax (e.g. binary octets, decimal digits, characters) of the DSP. The IDI identifies the network addressing domain and the network authority in that addressing domain responsible for allocating and ensuring unique value of the DSP.

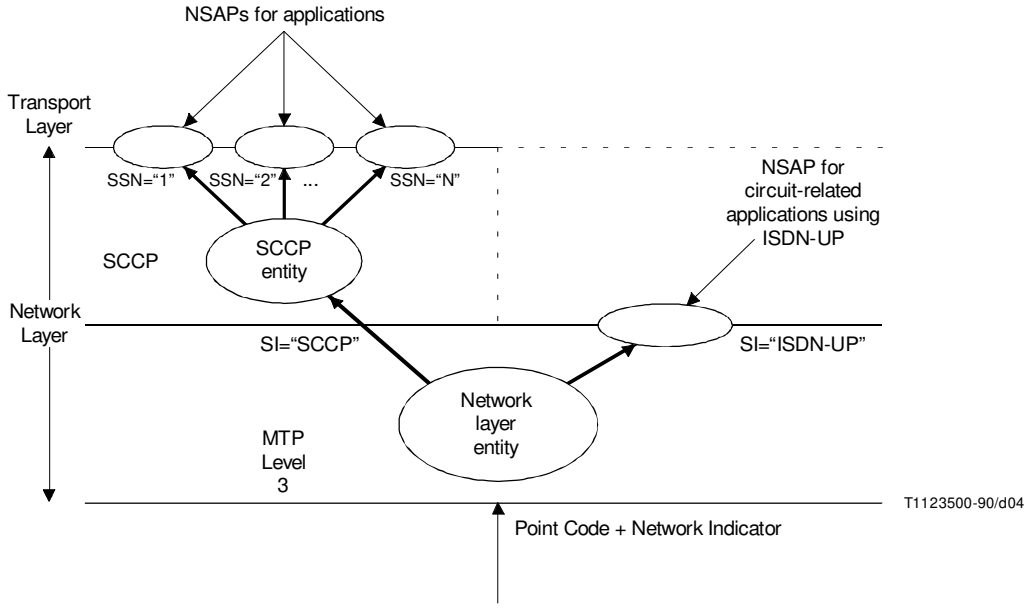
5.5 Lower Layer Addressing Relationships in International SS No. 7

The combination of SS No. 7 Point Code (PC), Service Information Octet (SIO) and SCCP Sub-System Number (SSN) meets the criteria for providing the semantics of an OSI Network (NSAP) address. It follows the hierarchical addressing structure defined for the OSI NSAP Address because CCITT defines the network addressing authorities who in turn define the addresses within their sub-domain. A part of the SS No. 7 PC, the 11-bit Signalling Area Network Code (SA/NC) field, together with the Network Indicator field in the SIO serves the purpose of the OSI NSAP ISP because they identify the sub-domain addressing authorities. Together with the remainder of the PC, which is domain-specific, an SS No. 7 node can be addressed in a "globally" unambiguous manner.

Within the Network at a node there are a number of NSAP Addresses with different types of upper-layer entries bound to these NSAPs. Once a node has been unambiguously identified, local selectors administered within that node identify the possible NSAPs. In certain cases, the SI field contains sufficient information to locate these NSAPs. CCITT has

standardized some of these SI fields. For instance, values have been assigned to those that directly access the ISDN-UP and TUP application entities of the call-processing application process. Also, another standardized SI locates another entity within the Network Layer which provides the functions of the SCCP. In this case, a further piece of addressing information, the SCCP SSN, is the local "selector" to distinguish between the possible NSAPs at the Network/ISP Layers' interface. CCITT has also standardized the values for a few SSNs to locate certain upper-layer entities, e.g. those for the OMAP (Operations, Maintenance and Administration Part) and the MAP (Mobile Application Part) AEs, but it need not do so in general. Communications between application processes at different nodes are ensured by the proper maintenance and administration of network directory functions like routing tables and global title translation tables.

These addressing relationships are illustrated in Figure 4.



NSAP	Network Service Access Point
SSN	Subsystem Number
SCCP	Signalling Connection Control Part
SI	Service Indicator
ISDN-UP	ISDN User Part
MTP	Message Transfer Part

FIGURE 4/Q.1400

Relationship of Point Code, Service Indicator and Sub-System Number to OSI NSAP

The NSAPs which are of particular interest are those that provide the interface to the SCCP-User for non-circuit related signalling applications. When Network Layer connections (NC) are set up between peer entities addressed through these NSAPs, they are identified by SCCP Local Reference Numbers which are, in Recommendation X.650 terminology, network-connection-endpoint-identifiers.

Connectionless data transfer using the N-UNITDATA primitive also occurs through these NSAPs though no connection-endpoint-identifiers are necessary in this instance.

The NSAPs for ISDN-UP and TUP directly locate, because of the absence of intervening layers, distinct Application Entity types belonging to the Call Processing Application Process(es).

5.6 Summary of Addressing Equivalents Noted for International SS No. 7

Table 1 shows the relationships between SS No. 7 and OSI Naming and Addressing terminology and functions that have been explored and noted in the previous subclause.

TABLE 1/Q.1400

Summary of Addressing Equivalencies for SS No. 7

OSI Term/Function	SS No. 7 Equivalent
Network Address	(PC, SIO, SSN)
NSAP IDP	(NI, SA/NC portion of PC)
NSAP DSP	(remainder PC, SI, SSN)
Network connection endpoint identifier	SCCP Local Reference Number

5.7 Further Study Item for Evolution of SS No. 7 Addressing

At present the SCCP Calling and Called Party Address, which consist of a Global Title and SSN, are provided to the upper layer in the N-CONNECT and N-UNITDATA indication primitives. The SCCP Global Title Translation function converts a title into a Network Address. In contrast, an OSI application directory function converts an Application Entity Title into a more detailed address of the form (PSAP, SSAP, TSAP, Network Address). The SCCP Global Title translation function in which a global title is converted into an NSAP Address of a node is simply the conversion of the generic name for a Network Address into an actual Network Address. In other words, the SS No. 7 GT currently is not providing an application Entity Title even though the Global Title is sent up to the upper layers in the N-UNITDATA and N-CONNECT indication primitives. Therefore, the SCCP Global Title and SSN do not provide the full OSI PSAP Address, but assume a one-to-one mapping with the more detailed address. Thus, if any greater distinction than is currently possible is to be made between upper-layer entities, it has to be performed by enhancing the current routing "directory" to provide the required upper-layer addressing information and enhancing the SS No. 7 protocol(s) to carry that information.

This last point requires consideration particularly in the case where it is desirable to locate further sub-structure in the Application Layer such as the Stage 2 Functional Entities within an Application Process at a physical node.

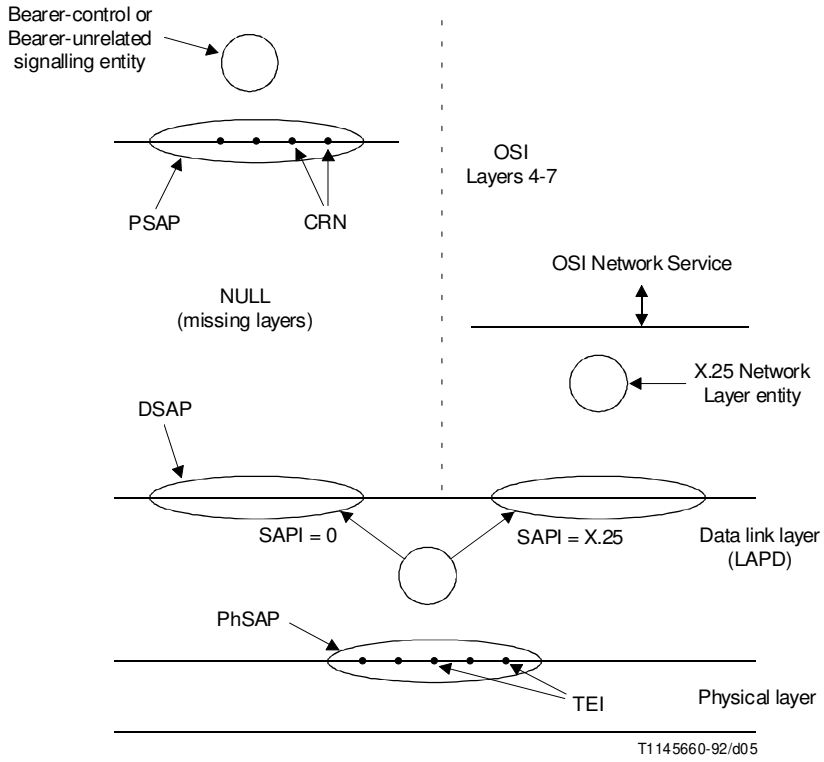
5.8 Addressing Equivalents for DSS 1

Elements of DSS 1 addressing information were discussed in 5.3. Figure 5 (DSS 1) parallels Figure 4 (SS No. 7).

DSS 1 provides in LAPD, Recommendations Q.920 and Q.921, procedures for broadcasting to and selecting from terminals connected to the interface. All messages from the user side on the interface are seen by the network side and by all other user entities on the interface. Similarly, all messages sent by the network side are seen by all the entities on the user side. Using the TEI and SAPI, the specific user entity and the specific capability of those it supports is identified, or the message is labelled as a "broadcast" message to all the user entities. User side Layer 2 entities ignore non-broadcast messages which do not match the assigned TEI and the SAPI. A specific instance of interaction is identified by the DSS 1 Call Reference above the Connection Endpoint Suffix and SAPI.

This means that all the users and the network element share a single Physical Service Access Point (PhSAP), with each entity (terminal or network element) being identified by a physical connection end-point identifier, called the Terminal End-point Identifier (TEI), at the PhSAP. Having accessed the Data Link entity via the PhSAP, the Service Access Point Identifier (SAPI) allows the choice between various Data link Service Access Points (DSAPs). A standardized value of 0 identifies the DSAP corresponding to the D-channel of the ISDN access, while the SAPI = "X.25" identifies the DSAP which offers the Recommendation X.25 Network Entity. Above the DSAP reached by SAPI = 0, there are no intermediate layer entities below the Application Layer. Each instance of the Application Layer entity which generates protocol for bearer related (see Recommendation Q.931) or bearer-unrelated (see Recommendation Q.932) signalling is identified by means of a Call Reference Number (CRN). The SAPI = "X.25" leads to a full OSI stack with its usual addressing.

There is no further addressing in DSS 1, therefore no further substructuring can be delineated.



PSAP Presentation Service Access point
SAPI Service Access Point Identifier
TEI Terminal Endpoint Identifier
PhSAP Physical Service Access Point
DSAP Data Link Service Access Point
CRN Call Reference Number

FIGURE 5/Q.1400

Application of OSI Addressing Concepts in DSS 1

6 Application of OSI Application Layer Concepts

6.1 Application of OSI Application Layer Concepts to SS No. 7

Given an Application Process (AP) static description, at some point in time an Application Process Invocation (API) will be created as a result of some stimulus outside the scope of this discussion.

When the API wishes to communicate, it will create an Application Entity Invocation (AEI) which is a collection of all the supportable ASEs that may be used in the communication.

Four aspects which are essential to every instance of communication must be understood. These are, in order:

- locating the application one wants to "talk" to (Addressing) (see 6.1.1);
- the "language" to be used to communicate (Presentation context) (see 6.1.2);
- the general "topic" of the "conversation" (Application context) (see 6.1.3); and
- the specific questions and responses related to that "topic" (operations, results and errors based on the ROSE framework) (see 6.1.4).

6.1.1 Locating the Remote Application

Addressing is the first essential aspect of communication which allows one AE to establish an association with another AE by first accessing the node where the peer AE is located (using the Network Address) and then following "an internal route" through the upper layer entities at that node (using the Transport, Session and Presentation selectors).

In OSI the initiating Application Entity consults a directory to locate the address of its peer. The initiator provides the directory with the Application Entity Title or the Application Process Title. The directory returns the appropriate Presentation Service Access Point (PSAP) which is a tuple = (P_selector, S_selector, T_selector, Network Address). The directory can provide this information to its users only if all applications register their address with the directory provided. Each part of a PSAP is inserted into protocol addressing information at the corresponding layers for routing to the peer AE (e.g. the Transport Layer protocol carries the Transport selector, etc.)

SS No. 7 applications use a similar directory technique. The application provides a Global Title to the "directory" – which is the SS No. 7 Global Title Translation function – which then provides the tuple = {SSN, PC}. The Point Code (PC) and SCCP Sub-System Number (SSN) are the addressing information that, because of the absence of intervening layers, effectively locates an AE-type at an SS No. 7 node. The SSN really selects an NSAP, but as there are no Transport, Session and Presentation Layer protocols in SS No. 7, the chosen NSAP is therefore "nailed-up" to a Presentation Service Access Point (PSAP). In OSI addressing, the PSAP locates an AE-type.

AE-types for standardized applications, e.g. OMAP, are accessed at every SS No. 7 node by distinct CCITT-standardized SSNs. Other SSNs, chosen locally at a node, are used to access AE-types which contain the application-specific communications protocols for other applications. These are managed by an SS No. 7 network operator to ensure an accurate and up-to-date routing "directory" which constitutes the Global Title Translation tables.

6.1.2 Determination and Uses of Abstract Syntax

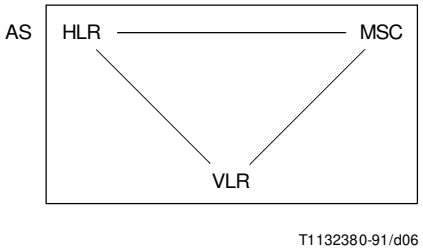
In OSI, the syntax of the Application Layer language to be used during an instance of communication is signalled during the setting up of the underlying Presentation Layer connection. This language is referred to as the Abstract Syntax. It is abstract because it is defined in terms of the structure and essential content of the information exchanged without preference or choice on how the information is actually encoded as "bits on the wire" for transfer to the peer entity via a communications medium. The actual encoding that is used is negotiated (or sometimes pre-determined) by the two peer presentation entities and is called the Transfer Syntax.

For example, Recommendation X.208 specifies ASN.1, a notation for specifying Abstract Syntaxes. Recommendation X.209, Basic Encoding Rules (BER), provides one possible set of encoding rules and thus represents a Transfer Syntax for Application Layer protocols that have been defined using ASN.1.

A presentation context is a mapping (for a period of time) between an abstract syntax and a transfer syntax. A presentation context is given an identifier to uniquely distinguish it from other presentation contexts in use on an association.

The initiating Application Entity invocation must include, in its request for an application association, the mandatory Application Context Name. This contains, among other things, the identification of the Abstract Syntaxes (AS) of the data structures to be used. Each of these ASs is given an identifier – a Presentation Context ID – by the Application Layer to which the Presentation Layer attaches a choice of Transfer Syntaxes. This information is carried in the Presentation protocol to the peer during the Presentation connection set up. The peer Presentation entity chooses one Transfer Syntax (from the offered choice) for each Abstract Syntax. The Presentation Context ID is used, during the subsequent data transfer phase, to transform the received PDUs into the appropriate Abstract Syntax before delivery to the appropriate protocol machine at the Application Layer where the AS can be meaningfully deciphered. Note that communicating AEIs must use identical abstract syntaxes.

Presently in SS No. 7 TC, there is only one abstract syntax which therefore requires unique operation codes at an AE addressed through a Network Address (PC + SSN) plus implicit P/S/T_selectors. This is illustrated by Figure 6, loosely based on the 1988 Recommendation Q.1051. Note that this requires that every node know the entire ASE even if it will not use all of it, and is a direct result of the way the 1988 MAP Recommendation Q.1051 is specified.



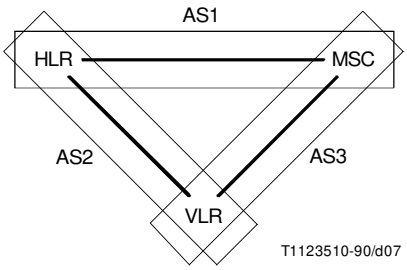
AS Abstract Syntax
HLR Home Location Register
VLR Visited Location Register
MSC Mobile Switching Centre

FIGURE 6/Q.1400

Recommendation Q.1051, MAP (1988)
Abstract Syntax as Illustrative Example

An alternative approach (purely to illustrate the use of Abstract Syntaxes) is illustrated in Figure 7. Here there are three contexts representing the interactions, e.g. between the MSC and the VLR. A unique Abstract Syntax may be specified for each of these. The operation codes may overlap across the contexts but must remain unique within the syntaxes. Other approaches are also possible.

SS No. 7 presently does not have any means to differentiate among multiple Abstract Syntaxes. As noted above, “large” ASEs (e.g. MAP) have been created as opposed to several smaller ASEs. We presently have an *a priori* Abstract Syntax and Transfer Syntax as the only alternative and which must be understood by both ends. There is no opportunity for negotiation and hence no need for a protocol to do negotiation. This situation is not expected to persist as discussion of the need for multiple Abstract Syntaxes and Transfer Syntaxes for ISCP and the Intelligent Network Application Part (INAP) are taking place.



ASn Abstract Syntax *n*, *n* = 1, 2, 3
HLR Home Location Register
VLR Visited Location Register
MSC Mobile Switching Centre

FIGURE 7/Q.1400

Alternate MAP Abstract Syntaxes as Illustrative Example

6.1.3 The Context for the Communications

Once the appropriate AE-type has been accessed, the "topic" of "discussion" between the peer application entity invocations is determined by the AC. The AC provides all the information necessary to create the appropriate Single Association Object (SAO) within an Application Entity Invocation (AEI). The SAO is a concise modelling description of all the communications functions of the AE-type that is required for this one instance of communication over an application association.

This is analogous to human interactions where one person broaches a "topic" of conversation which outlines the scope of the conversation but does not, as yet, determine the exact dialogue. That is dynamically generated by the speakers only if the subject is agreeable to both. Unlike humans, though, the OSI application association requires that the discussion does not stray from the topic.

An AE-type at a node can potentially support a very large number of communications capabilities for the Application Process. The AC serves the purpose of choosing from the available functions the specific functions needed for a single instance of communications. This leads to one further aspect of the explicit application association set up in OSI: the AC name is negotiated by the two peers, and the association fails if a mutually acceptable context cannot be agreed upon.

When an AE at an ISDN node supports a large number of capabilities, it is useful to signal up front just which of these capabilities are likely to be required during the instance of communications. In the case of SS No. 7 TCAP, logically related groups of TC-User defined remote operations are called TC-User ASEs. Each ASE is a collection of related operations that together provide some overall capability. In SS No. 7, the context for an instance of communications would consist of a list of such ASEs that might be used together with some rules on how they would be used in conjunction with each other (such as the sequencing of operations, which side can invoke which operation, etc.).

6.1.4 Specific Questions and Topics Within an Application Context

In OSI, first an association is established during which the Application Context is used to pull together the specific ASEs and the coordination rules governing their use. Only after this, service requests from the Application process give rise to appropriate Application Layer protocol exchange.

As there is an underlying Presentation connection supporting the association, the subsequent data exchange need only use the Presentation connection end-point identifier to reach the appropriate SAO while the Presentation Context Identifier serves to transform the incoming message encoding (Transfer Syntax) into the appropriate data structures (Abstract Syntax).

The remainder of this subclause describes the application-specific data exchange progressing through the use of the query/response paradigm. This is modelled as the invocation of Remote Operations and uses the ROSE protocol which is a key element in both DSS 1 and SS No. 7.

In SS No. 7, particularly in the case of 1988 TCAP Recommendations, the Transaction ID serves to identify the implicit association. As there is no protocol to support multiple Abstract Syntaxes, specific interactions necessary during a transaction are determined by a variety of means:

– unique operation codes;

– the same operation code but with different (optional) parameters;

– the same operation but with different parameter values.

These are options available to "route" the ROSE PDU to the appropriate User-ASE. These options may be thought of as a means of providing a "sub-context", i.e. refining the initial Application Context, which still governs the overall "topic of conversation".

Thus, in TC, the effect at the start of a communication is as illustrated in Figure 8.

Considering the ISO ALS specification, and a Single Association Object in it, and relating this to TC, Figure 9 may be drawn. Observe that the TR-BEGIN request may be considered equivalent in some ways to a P/S/T-CONNECT request with the User Data field containing the (presently implicit since there are no alternatives to be indicated) association protocol plus other ASE protocols.

Unlike OSI, ISDN applications do not have an explicit association establishment procedure that must be completed before data may be exchanged. For signalling efficiency, data is transferred at the time of the set-up, i.e. with the first message exchanged.

Thus, in SS No. 7 TCAP applications where remote operations are embedded within the "association/transaction set-up" request, it is assumed that the two AE invocations know the AC beforehand. One way to mimic the OSI association establishment procedure, though at the cost of having additional messages, is to use an "empty" BEGIN message containing the proposed AC to which the response is an empty CONTINUE containing the acceptable AC, after which the exchange of remote operations proceeds. Without an explicit "association" establishment procedure, the AC, as defined in OSI, has no clear meaning.

6.2 Association Control Requirements for Signalling

The requirement of establishing an association before the transfer of any Application Layer PDUs does not suit applications requiring stringent real time performance. Therefore, there is the need for an "efficient" association control protocol.

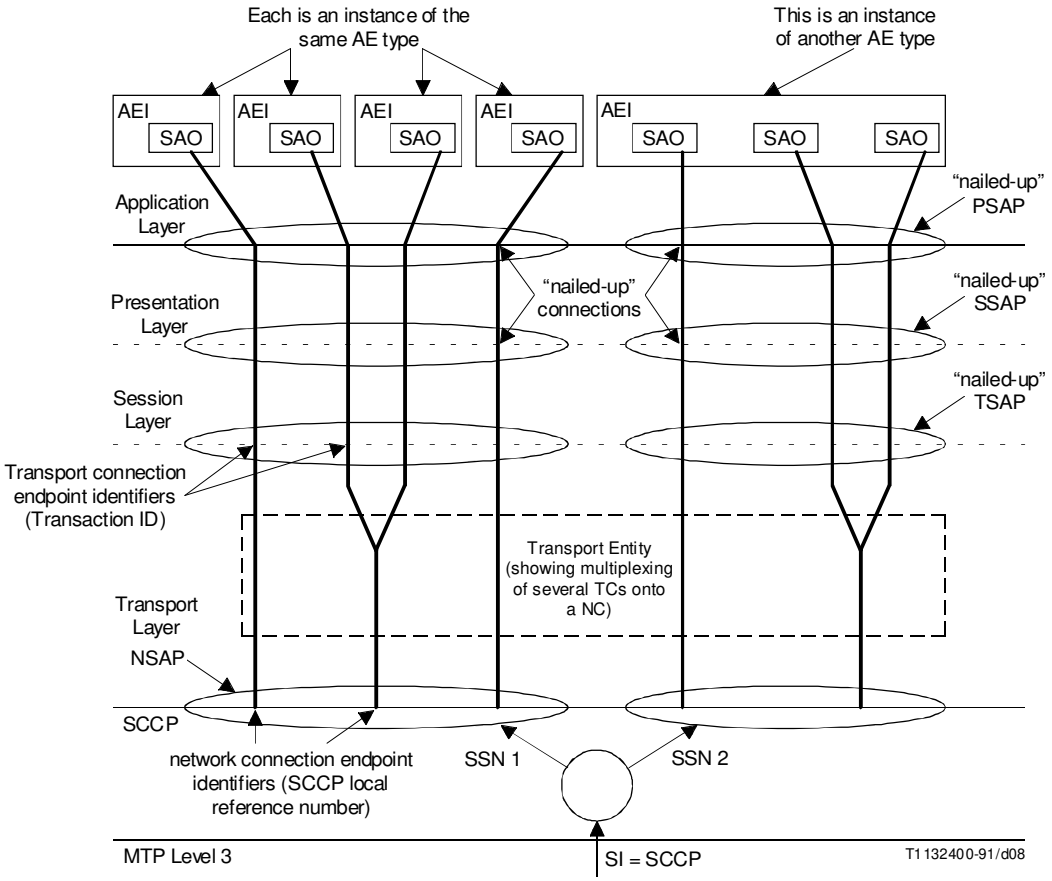
OSI has defined an ACSE but this is considered unsuitable for signalling needs for efficiency reasons. The following list of desirable characteristics is identified for signalling association control:

– support of unconfirmed and pre-arranged termination;

– simplicity at establishment; and

– allow ROSE (and possibly other) PDUs to be carried within the establishment PDU.

The next two subclauses review the essential elements of the Association Control Service Element (ACSE) and are intended to convey an understanding of the essential elements of this protocol. More detailed information should be obtained through consulting the indicated references.



AE	Application Entity
AEI	Application Entity Invocation
SAO	Single Association Object
PSAP	Presentation Service Access Point
SSAP	Session Service Access Point
TSAP	Transport Service Access Point
NSAP	Network Service Access Point
TC	Transport Connection
NC	Network Connection
SCCP	Signalling Connection Control Part
SSN	Subsystem Number
MTP	Message Transfer Part
SI	Service Indicator

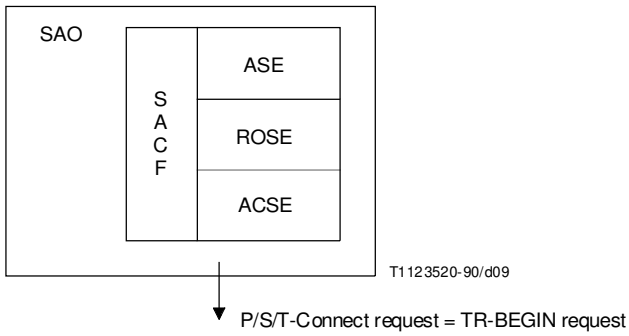
FIGURE 8/Q.1400

Situation in SS No. 7

6.2.1 Connectionless ACSE

ISO/IEC 8649/DAD2 and ISO/IEC 10035 contain the ACSE service and protocol descriptions respectively for the connectionless mode.

The connectionless ACSE service is a non-confirmed service and makes use of the A-UNITDATA primitive and PDU. The connectionless ACSE uses the presentation connectionless service: P-UNITDATA. During an instance of communication, the existence of both the sending and receiving AEIs is presumed. (In Table 2, M is mandatory, P is subject to conditions defined in ISO/IEC 8649/DAD2, and = means that the value in the indication must be identical to that in the request.)



SAO Single Association Object
SACF Single Association Control Function
ASE Application Service Element
ROSE Remote Operations Service Element
ACSE Association Control Service Element

FIGURE 9/Q.1400

Relating ISO ALS and SS No. 7 TC

TABLE 2/Q.1400

Selected A-UNITDATA Parameters

Parameter Name	Req.	Ind.
Application Context Name	M	M(=)
User Information	M	M(=)
Quality of Service	P	
Presentation Requirements	P	P

An A-UNITDATA procedure is directly related to that defined for the P-UNITDATA service. The requestor issues an A-UNITDATA request primitive. Use of the service is restricted to connectionless operation. The ACSE service provider issues an A-UNITDATA indication primitive to the acceptor. No response primitive is returned. Two AEIs simultaneously issuing A-UNITDATA requests to each other does not result in a collision and both A-UNITDATA indications are accepted. If the receiving Association Control Protocol Machine (ACPM) finds any parameter unacceptable, the PDU is discarded.

A subset of connectionless ACSE version 1 corresponding to Table 2 may be defined as follows:

```
Connectionless-ACSE-1S
{ ccitt recommendation q 1400 modules(0) cl-acse-1s(0) version1(0) }
DEFINITIONS ::=
-- Connectionless-ACSE-1S represents a subset of Connectionless-ACSE version 1
BEGIN
AUDT-APDU ::= [APPLICATION 0] IMPLICIT SEQUENCE
        { protocol-version [0] IMPLICIT BIT STRING { version1(0) }
                        DEFAULT version 1,
          application-context-name [1] Application-context-name,
          user-information [30] IMPLICIT SEQUENCE OF EXTERNAL
}
END
```

The User Data field of the Application Association Request (AARQ) and the Application Association Response (AARE) APDUs of the connection-oriented ACSE were not intended to carry other Application Layer protocols such as the ROSE PDU. Clearly, such an approach is not appropriate for the connectionless ACSE PDU.

The requirement of establishing an association before the transfer of any Application Layer PDUs does not suit applications requiring stringent real time performance. Therefore, there is a need for an "efficient" association control protocol.

It should be noted that the current TC-UNI service is the SS No. 7 equivalent service to the A-UNITDATA service. Neither the A-UNITDATA nor TC-UNI services are suitable to control the exchange of several related signalling messages. However, they can be used to control the transfer of signalling information such as alarm reporting.

If enhancements to the TC-UNI are required, they should be in line with the A-UNITDATA service.

6.2.2 Connection Oriented ACSE

Table 2/X.217 lists thirty-one parameters for A-ASSOCIATE. The most significant ones are shown in Table 3. Other parameters not shown include calling and called application process titles and application entity titles, qualifiers and invocation identifiers, and various parameters related to use of presentation services. All of the parameters omitted are optional or applicable to specific defined cases.

TABLE 3/Q.1400

Selected A-ASSOCIATE Parameters

Parameter Name	Req.	Ind.	Resp.	Conf.
Mode	U	M(=)		
Application Context Name	M	M(=)	M	M(=)
User Information	U	C(=)	U	C(=)
Result			M	M(=)
Result Source				M
Diagnostic			U	C(=)
Quality of Service	P	P	P	P
Presentation Requirements	P	P	P	P
Session Requirements	P	P	P	P

C Conditional
M Mandatory
P Subject to conditions defined in Recommendation X.216
U User option

The Mode is either "normal" (default) or "X.410-1984". The latter case is defined explicitly for 1984 X.410 MHS using RTSE.

The Application Context Name proposes a context. The acceptor returns the same name or proposes an alternative. If the alternative is not acceptable, an A-ABORT may be issued.

User Information is dependent on the application context.

The Result indicates the result of using the A-ASSOCIATE service and is one of "accepted", "rejected (permanent)" or "rejected (transient)".

The Result-Source indicates the source of the Result and Diagnostic parameters. It is one of "ACSE service-user", "ACSE service provider", or "presentation service provider".

The Diagnostic is used if the Result is "rejected" (permanent or transient).

Quality of Service, Presentation and Session Requirements are defined in Recommendation X.216.

Table 3/X.217 lists three parameters for A-RELEASE as shown in Table 4.

TABLE 4/Q.1400

A-RELEASE Parameters

Parameter Name	Req.	Ind.	Resp.	Conf.
Reason	U	C(=)	U	C(=)
User Information	U	C(=)	U	C(=)
Result			M	M(=)

Reason is "normal", "urgent", or user-defined. Result is "affirmative" or "negative".

The structured dialogue handling capabilities of TCAP are modelled on the CO-ACSE (connection-oriented ACSE) services.

An A-ABORT is a user abort. A-ABORT parameters are shown in Table 5.

TABLE 5/Q.1400

A-ABORT Parameters

Parameter Name	Req.	Ind.
Abort Source		M
User Information	U	C(=)

Abort source is used to indicate the initiating source of the abort: ACSE service user or ACSE service provider.

A-P-ABORT indicates an abort by the presentation service. A-P-ABORT parameters are shown in Table 6.

The Provider Reason is the indication of why the Provider is aborting.

TABLE 6/Q.1400

A-P-ABORT Parameters

Parameter Name	Ind.
Provider Reason	P

6.3 ROSE

Recommendation Q.775 contains guidance on the usage of TCAP. The information contained in Recommendation Q.775 is useful in understanding ROSE and since ROSE will be widely used in the network and access protocols (e.g. TCAP, Facility IE, etc.).

The reader's attention is drawn to the OPERATION and ERROR macros in Recommendation X.229 and also discussed in Recommendation Q.775.

7 Management Functionality

Recommendation X.700, OSI Management Framework, provides general principles for the management framework. This clause describes briefly how this applies to SS No. 7. Recommendation Q.940 describes the application to DSS 1. For a more detailed discussion, refer to TMN.

Figure 10 shows that each entity in the protocol architecture has a Level Management Entity (LME) associated with it. The LME may be more or less explicitly defined within the level. For example, in the SCCP, there are a number of management functions (e.g. SSN management) which constitute the LME for the SCCP.

All the entities in the architecture communicate with a Management Information Base (MIB) through a Level Management Interface (LMI). The MIB is used by the System Management Application Process [SMAP, of which the Operations and Administration Application process (OMAP) is a part] in performing OA&M functions. The Application Management Interface (AMI) represents the interaction between SMAP and the other application processes at the node that SMAP supports.

8 Layer 4, 5, 6 Guidelines

8.1 General

This subclause provides general guidelines relating to Layers 4, 5 and 6 (Transport, Session and Presentation) of the OSI model from a signalling perspective.

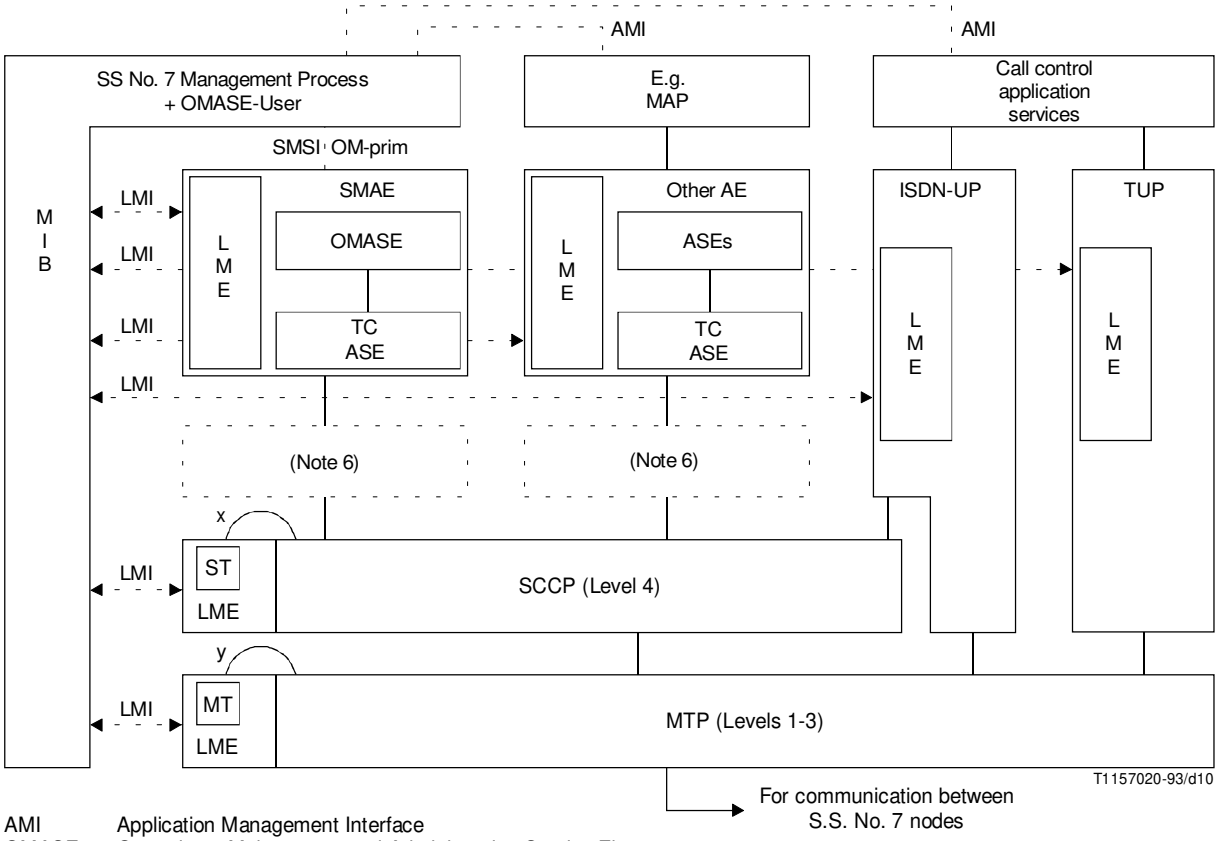
8.2 Layer 6 – Presentation

This subclause provides information on the Presentation Layer services and protocol based on Recommendations X.216 and X.226. This subclause rationalizes the use of the Presentation Layer in the telephony network and provides guidelines on its usage.

At present, SS No. 7 TCAP employs a Presentation Layer function which is the representation transformation of X.209-encoded PDUs into a single Abstract Syntax.

8.2.1 Use of the Presentation Context in OSI

The information content and structure of Application Layer PDUs are specified in OSI in Abstract Syntax using the ASN.1 notation defined in Recommendation X.208. For two Application Entity Invocations (AEIs) to communicate successfully over an application association, they must have an agreement on the set of Abstract Syntaxes that will be used, i.e. on the sets of APDUs that will be exchanged over the given association. Application Layer information defined in some, possibly several, abstract syntaxes is handed over to the Presentation Layer as a series of Presentation Data Values (PDVs). Each PDV must belong to a single named Abstract Syntax.



AMI Application Management Interface
OMASE Operations, Maintenance and Administration Service Element
SMSI Systems Management Service Interface
LMI Level Management Interface
MIB Management Information Base
LME Level Management Entity
ST SCCP Tester
MT MTP Tester
OM Operation and Maintenance

NOTES

1 Dotted lines (but not boxes) denote direct management interfaces. Only the SMSI (see Note 5) is realized with primitives.

2 The LMI (Level Management Interface) is not a subject for standardization.

3 The AMI (Application Management Interface) is not a subject for standardization.

4 The items managed by OMAP can be regarded as conceptually resident in the MIB.

5 The SMSI is the systems management service Interface; the OM primitives are defined for use over it for managed object functions defined in Recommendation Q.753.

6 OSI Layers 4, 5 and 6 are null in SS No. 7. TC forms the bottom of OSI Layer 7, SCCP the top of OSI Laser 3 (but is in SS No. 7 level 4).

7 Interface x uses subsystem numbers to test the SCCP using the SCCP Test (ST), interface y uses SIO to test the MTP using the MTP Tester (MT).

8 The LME (Level Management Entity) is defined for management of and within each level of SS No. 7. This is conceptually where each managed item resides as far as the level is concerned.

FIGURE 10/Q.1400

SS No. 7 Management and Internal Configuration of an SP

It is the function of the Presentation Layer to transfer unchanged these PDUs using some mutually acceptable concrete encoding scheme, or possibly schemes. It is the responsibility of the initiating AEI, during the establishment of communications, to inform the Presentation Layer of the ASs that will be used during the subsequent communications over the application association. Knowing the ASs to be used, it is the function of the Presentation Layer to negotiate, during the presentation connection establishment phase, a suitable Transfer Syntax (TS) for each AS. The pairing of an AS with a mutually acceptable TS is defined as a Presentation Context and is identified by a unique integer value – the Presentation Context Identifier – during the lifetime of the application association and its underlying presentation connection. During the application data transfer phase, incoming messages are converted using the Presentation Context ID from their TS into the appropriate AS for delivery to the appropriate ASE. Figures 11, 12 and 13 provide a pictorial description of these Presentation Layer functions.

In Figure 11, the initiating AEI provides an identifier (the Presentation Context ID) for each of the three ASs that it anticipates using during communications. The initiating Presentation Service provider proposes, for each AS named by its user, a list (in general) of TSs that it is capable of supporting for that Presentation Context. The application establishment PDU is transferred within the presentation-connection-establishment PDU which carries the list of proposed Presentation Contexts. On receiving the presentation-connection-request PDU containing this information, the peer presentation entity informs its user of those (if any) ASs whose representation transformation it cannot support with the proposed TSs. (It is assumed, for simplicity, that such is not the case in this example.)

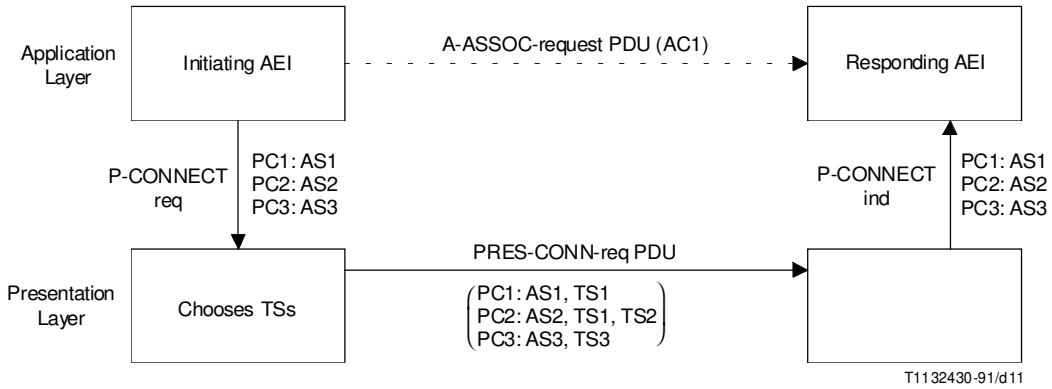


FIGURE 11/Q.1400

Presentation Connection Request

Figure 12 shows the case where the responding AEI accepts the application association with the proposed ASs. The responding Presentation entity chooses, for each AS in the Presentation Context list, one TS from those proposed by the initiator for that Presentation Context. The set of all such pairings of each AS with a TS is called the Defined Context Set (DCS). The DCS remains in effect for the lifetime of the presentation connection. (For simplicity, and because the need does not appear to exist for signalling applications at the moment, the optional possibility available in the OSI Presentation Service of altering the Presentation Contexts during the lifetime of the presentation connection is omitted from this discussion.)

In Figure 13, the user requests the transfer of some APDUs which it provides to the Presentation entity as a series of PDUs. Each Presentation Data Value (PDV) is marked with its Presentation Context ID which permits the sending Presentation entity to transform the AS into the correct TS and the receiving Presentation entity to transform the TS into the correct AS.

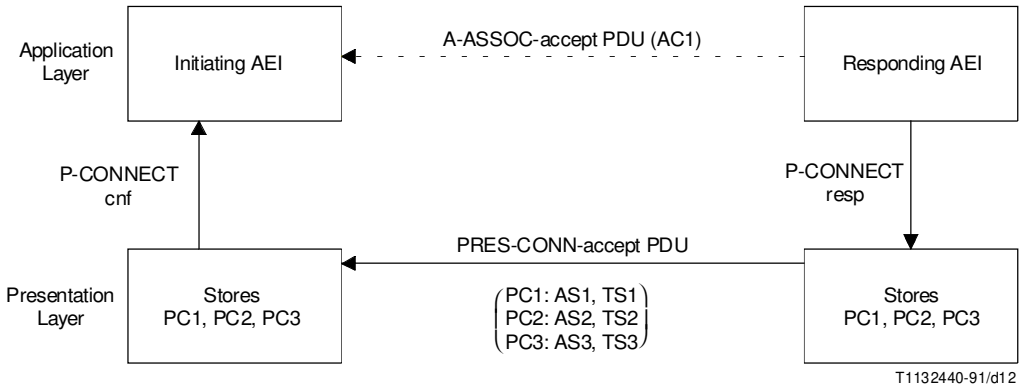
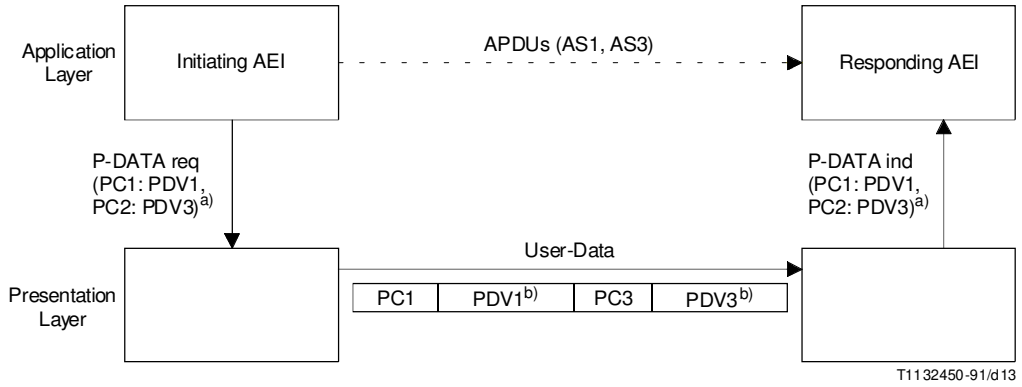


FIGURE 12/Q.1400

Presentation Context Acceptance



a) Each PDV is marked with its Presentation Context ID and is in some local (implementation dependent) representation of the appropriate abstract syntax.

b) Each PDV is encoded according to the negotiated Transfer Syntax for that Presentation Context.

FIGURE 13/Q.1400

Presentation Data Transfer

It is possible to have APDU exchange using a default Presentation Context. If so, the above method is simplified. The initiating AEI provides a single AS for which the Presentation entity provides a TS. This information is conveyed during the presentation connection establishment and the AS and TS has to be acceptable to, respectively, the responding AEI and Presentation entity.

It is also possible to have an implicit default Presentation Context if the AS to be used is known *a priori* to the initiating and responding AEIs and the default TS is known as acceptable to the Presentation service provider. No information on the Presentation Context is conveyed in protocol. This is the current situation in SS No. 7 where both AEIs in communication know the TS in use (which is Recommendation X.209 for TCAP and an octet-aligned, bit-oriented encoding defined in Recommendation Q.763 for the ISDN User Part) and any AEI supports only a single, unnamed AS. This is possible because each of these types of SS No. 7 applications are located at separate addresses (different Sub-system Numbers, and/or Service Information Octet values) so that the location serves to identify the default Presentation Context. Indeed, it is the limitations of the present situation – restriction to a single, default Abstract Syntax – that have led to this discussion.

8.2.2 Some Aspects of the Presentation Protocol

The OSI presentation protocol is defined in Recommendation X.226. The full Presentation protocol for presentation connection establishment/release is not described here as it is not directly relevant to the proposed needs. The portions of the Presentation connection establishment protocol that are relevant are:

```
Presentation-context-definition-list ::= SEQUENCE OF SEQUENCE {
        Presentation-context-identifier,
        Abstract-syntax-name,
        SEQUENCE OF Transfer-syntax-name }
Default-context-name ::= SEQUENCE       { [0] IMPLICIT Abstract-syntax-name,
                                          [1] IMPLICIT Transfer-syntax-name }

where,

Transfer-syntax-name ::= OBJECT IDENTIFIER
Abstract-syntax-name ::= OBJECT IDENTIFIER
Presentation-context-identifier ::= INTEGER
```

The Presentation-context-definition-list identifies each AS by an identifier and proposes a choice (in general) of Transfer Syntaxes for encoding the data belonging to this AS.

After the presentation-connection set-up, the APDUs are transmitted as a series of PDVs, with each PDV differentiated by a Presentation Context ID, in the User Data field of the Presentation Data PPDU (TD PPDU). The Presentation data transfer protocol is defined in Recommendation X.226 as:

```
User-Data ::= CHOICE    { [APPLICATION 0] IMPLICIT Simply-encoded-data,
                          [APPLICATION 1] IMPLICIT Fully-encoded-data }
Simply-encoded-data ::= OCTET STRING
Fully-encoded-data ::= SEQUENCE OF PDV-list
PDV-list ::= SEQUENCE {
                        Transfer-syntax-name OPTIONAL,
                        Presentation-context-identifier,
                        Presentation-data-values CHOICE
                          { single-ASN.1-type [0] ANY,
                            octet-aligned       [1] IMPLICIT OCTET STRING,
                            arbitrary           [2] IMPLICIT BIT STRING }
                        }
```

The Simply-encoded-data type is used when a default Presentation Context is used, i.e. both peers have either an explicit or implicit knowledge of the AS and the TS being used during communications, or when the DCS has only one element (with no option of changing it during the lifetime of the connection). One caveat when using this type is the requirement that the TS chosen to encode concatenated PDUs (all belonging to the same and only AS) be self-delimiting i.e. knowing where one information element ends and another begins. [The BER (of Recommendation X.209) is one example of a self-delimiting encoding scheme.]

The Fully-encoded-data is used when the DCS contains more than one Presentation Context. (Clearly the default context is not in use.) Consider a series of PDUs (each marked with its Presentation Context) given to the Presentation Layer to encode. For a given Presentation Context, if a PDV is a single ASN.1 type, it is encoded by the BER using the option "single-ASN.1-type". If the PDV belonging to this Presentation Context is not a single ASN.1 type and the TS chosen results in an integral number of octets, the "octet-aligned" option is used. If neither of these hold, the third option "arbitrary" is used, with the requirement that the transfer syntax chosen be self-delimiting.

8.2.3 Further Study Items for Application of Presentation Layer Concepts

In SS No. 7, each Presentation Context is implied by the encoding of the incoming message. This presently applies to TCAP applications where there is a well defined AS (given in Recommendation Q.773) as well as encoding. For any AE containing the TCAP ASE together with any TC-User ASEs, there is only one Abstract Syntax. [This is the motive behind ensuring the uniqueness of operation and error codes as outlined in Recommendation Q.775. If there was more than one AS, operation and error codes would need to be unique only within the scope of a particular Abstract Syntax (see Recommendation X.219); but it would be necessary to find a way to identify each AS.]

The use of more than one AS would alleviate a "language" problem concerning whether to import/export operation and codes between ASN.1 modules by type or value. A common problem that arises when ROSE/TCAP-User ASEs are defined independently is to find a way to ensure that, should the occasion arise, several ASEs can coexist in one AE-Invocation. If the ASEs have been defined such that the constituent operation and error codes have already been assigned "local" (i.e. integer) values, then the assignment of a separate AS to each ASE would alleviate the "language" problem that could arise if operation and error codes in different ASEs had the same value. Each ASE is identified by a separate Presentation Context Id.

It is also noted that global operation and error code values (i.e. object identifiers) can solve the above-mentioned issue. The use of the OBJECT IDENTIFIER requires several octets to encode (vs. at least one for integer operation and error codes).

The most constraining way of ensuring uniqueness of local operation/error codes is to assign a range of integer values to each ASE that is identified. Of course, not all values in that range need to be used. This requires coordination of range boundaries and may not be practicable if ASEs are borrowed from other non-signalling standards.

8.3 Layer 5 – Session

This subclause is a placeholder for material which will describe the Session Layer service and protocol based on Recommendations X.215 and X.225 and will identify the functional units suitable for a telephony network when requirements are identified.

8.4 Layer 4 – Transport

This subclause is a placeholder for material which will describe the Transport Layer service and protocol based on Recommendations X.214 and X.224 and will identify the classes of service suitable for telephony network when requirements are identified.

The exchange of several related signalling messages over a connectionless network service requires the existence of an explicit end-to-end relation. Except if a specific mechanism is defined within the application (e.g. circuit identification in TUP), this relation has to be established by the exchange of explicit local references.

This is one of the basic transport functions in the OSI environment: in SS No. 7, this function is currently provided by the TCAP TSL. However, it should be noted that the TCAP TSL does not provide any other transport specific services such as multiplexing or segmenting.

It is for further study whether the Transport Protocol may provide the basis for any further proposals for enhancements to the TCAP Transaction Sub-Layer. ISO has specified the use of Transport Protocol Class 4 over a connectionless mode Network Service. This protocol is "heavyweight" as connectionless data networks have minimal Quality of Service requirements. It will be useful to investigate the possibility of using a simpler transport protocol over CL-SCCP as the latter has extremely robust characteristics.

9 Layer 1, 2, 3 Guidelines

Recommendation Q.700 contains material relevant to the Layer 1, 2 and 3 services available from SS No. 7. Recommendations in the X.200-Series provide related information on services provided by these layers. Recommendations Q.701 through Q.704 specify the MTP levels 1 to 3. Recommendations Q.711 through Q.714 specify the SCCP. Recommendations I.430 and I.431 specify the DSS 1 Physical Layer, while Recommendation Q.921 specifies the LAPD.

10 Convergence Functions

A Convergence Function is one that is intended to replace functions of missing OSI layers or sublayers, or interface to non-OSI layers such that the user of the Convergence Function can behave as if the Convergence Function were providing the services of the applicable layer of the OSI Reference Model. The principal effect of adding a Convergence Function is that it transfers protocol control information to perform a function not performed by the underlying layer(s), to provide the expected OSI layer service to its user. In general, missing functions should be provided by incorporating appropriate subsets (e.g. protocol classes, FUs) of the relevant OSI layer protocol instead of re-specifying the functions as part of a Convergence Function. For example, the function which would map Network Layer service primitives to a specific sub-network (e.g. SS No. 7 SCCP) service primitives is called a Convergence Function.

SS No. 7 applications using TCAP (i.e. the TCAP-User plus TCAP represent an Application Layer protocol), and the ISCP (whose structure is based on ISO's ALS), are based on use of the SS No. 7 SCCP. The SCCP provides an OSI-like Network Layer service. Both the ISCP and TCAP applications utilize the connectionless services of the SCCP. To maximize the portability of these applications to any other sub-network, they need to be specified so as to make use of an OSI-like Network Layer service.

11 Applying Protocol Architecture Guidelines: Intelligent Network Application Part (INAP)

This clause describes how the protocol architecture guidelines are applied to the Intelligent Network Application Part. The details of the INAP specification may be found in Recommendation Q.1218.

11.1 How IN Concepts are Realized in Protocol

In order to aid the definition of IN capabilities, the CCITT has adopted a method of specification – called the IN Conceptual Model – where one proceeds from the highest level of abstraction (e.g. what a service provides to the end-user) to the lowest level of abstraction, viz. the details of the protocol between network elements that realize the service. An aid to this is the concept of "planes", of which three are relevant to this discussion, which allows the IN to be viewed according to:

- the overall capabilities that have to be provided by a network to design a service for the end-customer (the Global Functional Plane);
- the effects of distribution of various capabilities within a network in achieving the overall effects (the Distributed Functional Plane); and
- the physical realization of these capabilities through standardized protocol between network nodes (the Physical Plane).

This "top-down" approach for building service-independent IN capabilities, which can be accessed through standardized interfaces, is discussed in the subclauses below. Only those features of each "Plane" that are directly relevant to an understanding of the INAP are discussed below.

11.1.1 View from the Global Functional Plane

The Global Function Plane (GFP) provides a global, abstract view of the capabilities provided by the network as perceived by its service designers.

To this end, the IN standards have defined a basic unit of modularity called the Service Independent Building-block (SIB). SIBs are defined to be abstractions of service-independent network resources/functions that are visible and accessible through standardized interfaces. Each SIB represents a particular network capability realized through a (related) collection of procedures. For example, the ability of a network element to interact with an end-user can be modelled as a SIB. The procedures that make up such a SIB are, for instance, "play an announcement", "collect user-entered digits", etc. The SIBs are manipulated via their standardized interfaces by Service Logic Programs (SLPs) to provide service-specific functionality. As the Global Functional Plane (GFP) "hides" the effects of the distribution of capabilities, the SIBs are seen as "monolithic" entities by "users" of the SIBs.

Examples of SIBs are:

SIB: TRANSLATION

Procedures: Translate [a set of digits], …

SIB: USER INTERACTION

Procedures: GivePrompt, CollectDigits, …

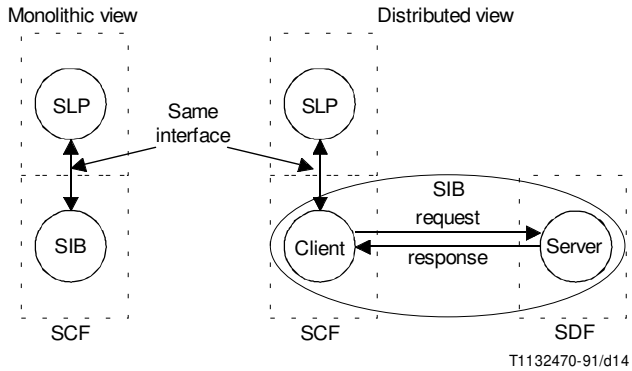
In the above examples, the capabilities of a SIB such as "Translation" can be used in many different situations: for example, such a SIB can be invoked by a Freephone service program to translate the Freephone number to a network-routable number; for another service, the same SIB can be invoked to translate a calling number into the caller's name.

11.1.2 View from the Distributed Functional Plane

An IN is not in reality a monolithic entity as perceived from the viewpoint of the GFP, but rather a distributed configuration of physical entities as viewed in the Physical Plane of the IN Conceptual Model. The SIBs which are perceived as single monolithic entities in the GFP – for purposes of service design – may actually be realized from a distributed configuration of capabilities implemented in different Physical Entities within a network. Thus protocol will be needed to correlate the actions of the distributed configuration of capabilities. To facilitate the definition of the physical entity capabilities and protocol needed to realize each SIB, a reference model has been defined in the Distributed Functional Plane (DFP) (similar to the "3-Stage Method, Step 2.1 Functional Model"). This model defines a number of Functional Entity types (FEs), e.g. Service Switching Function (SSF), Service Control Function (SCF), Service Resource Function (SRF) and Service Data Function (SDF) which are described in the paragraph below. A Functional Entity is a grouping of various service-providing capabilities. It is intended to be substantially generic so that it can be used as a template for modelling the distribution of all IN-based service features and the SIBs. Each FE represents a grouping of capabilities which must be realized within a single physical entity.

The SSF contains the functions beyond those needed for basic call control (i.e. classical call processing at a telephone exchange) that access new supplementary services as provided by the SCF. The SCF contains IN service logic for handling service-related call processing activities. When certain decision points are reached during the processing of a call, the SCF provides the service-dependent information which allows call processing at the SSF to progress. The SDF contains functions that handle access to and management of network- or service-related data. The SRF provides a number of resources (e.g. speech recognition devices, DTMF tone receivers and generators, synthesized speech processing devices, etc.) which can be accessed by other network elements.

A (monolithic) SIB mapped onto the DFP is decomposed into an interacting set of capabilities which can be modelled as a client-server pair, with each partner located in the different Functional Entity types. This is illustrated in Figure 14.



SLP Service Logic Program
SIB Service Independent Building Block
SCF Service Control Function
SDF Service Data Function

FIGURE 14/Q.1400

SIB Decomposition as a Client-Server Pair

Clients and Servers have the following well-defined meaning in object-oriented design: an object provides access to its capabilities through a (set of) well-defined interface(s). It is possible that the services provided by this object are, in reality, provided by an internal decomposition of this object into a group of objects that interact to provide the resultant external behaviour. In one particularly useful decomposition called the client-server model, two interacting objects can be defined such that one object – called the client – makes use of the services provided by another object – the server. It does it through the exchange of a set of defined "messages"/"information flows"/"operations" also called protocol. The client-server interface is the set of operations that the client object invokes on a server object to provide a service for the encapsulating "outer" object so that the latter's resultant interface is unchanged.

As shown on the left-hand side of Figure 14, the interface between the Service Logic Program (SLP) and the SIB is defined in the GFP. This interface is preserved despite the fact that it is convenient in the distributed view (right-hand side of Figure 14) to show the distribution of the SIB's capability as a client-server pair.

For example, a Translation SIB (performing, for example, a number translation procedure) can be modelled as being (partially) realized in both a Service Control Function (SCF) and Service Data Function (SDF), with the translation requestor (or "client") in the SCF and the translator (or "server") in the SDF.

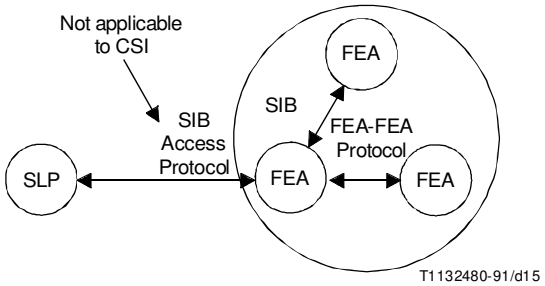
As shown on the left-hand side of Figure 14, the service logic interacting with a SIB via a standardized interface has the impression that it is interacting with the entire SIB. In a distributed realization, as shown on the right-hand side of Figure 14 it is interacting with only the "client" that is implemented in the SCF. The "client" cooperates with its "server" in the SDF, as necessary, to complete the procedure requested. In this particular example, virtually all the processing associated with the Translation SIB would be done by the "server" in the SDF, with the "client" in the SCF providing little more than remote access to its partner in the SDF.

Likewise, a User Interaction SIB can be modelled as an SCF-Specialized Resource Function (SRF) client-server pair, with the "client" in the SCF providing remote access to the "server" in the SRF, and prompting the latter to perform the actions appropriate to user interaction, such as playing some announcement and collecting the digits entered by the user.

Thus it is necessary to define the information flow(s) between the "client" and "server" to ensure that they transparently perform the function requested of the SIB.

As all the SIBs will be mapped onto the same model in the DFP, each FE will be allocated clients/servers from different SIBs. For example, an SSF will contain "servers" from the decomposition of the Basic Call, Charging and Status Notification SIBs. A description of all the SIBs may be found in Recommendation Q.1213, "Global Functional Plane for the Intelligent Network capability Set 1." A SCF will contain a client for the "Connection" procedure of the Basic Call SIB and a server for the "Translate" procedure of the Translation SIB. The total set of information flows between any two FEs will be the number of client/server information flows that they support.

A similar view is provided by Figure 15 with respect to the FEA to FEA protocol and the "SIB Access Protocol", which is apparently like the "Application Programming Interface" or API under discussion in the studies on Intelligent Networks. It is understood that the latter will not be specified for CS1 but is recognized as an essential portion of the evolving set of IN specifications.



SLP Service Logic Program
SIB Service Independent Building Block
FEA Functional Entity Action

FIGURE 15/Q.1400

SIB Access and FEA-FEA Protocol as Visible in the DFP

11.1.3 View from the Physical Plane

As the name suggests, the Physical Plane is where the abstractions of the GFP and the DFP are realized by allocating the Functional Entities to the various network elements, or, as they are called, Physical Entities. The relationship of the SIBs and the FEs to the INAP is subtle, because it must be remembered that the SIB is defined in the GFP. The SIB "casts a shadow" of its presence in the DFP, where each SIB's shadow falls on one or more FEs.

In the previous subclause it was shown that each SIB can be modelled in the DFP as a family of client/server pairs, whose behaviour is correlated by information flows. A group of clients or servers from different SIBs can be allocated to a particular Functional Entity (FE). A complete FE must be implemented in one Physical Entity. However, different FEs can be implemented in the same or different Physical Entities. If two FEs are in remote physical entities connected by a network, the information flows defined in the DFP are realized in the Physical Plane by standardized OSI-based Application Layer protocol, i.e. an "IN Application Protocol (INAP)". If two FEs are in the same physical equipment, the realization of their information flows is a local matter and not subject to standardization. However, since the distribution of the FEs to the PEs is not determined beforehand, all the information flows among FEs need to be realized as a part of the INAP. Also, not every network element has to implement the capabilities of each and every SIB that is defined; if a SIB is not implemented in the network, then the corresponding client-server pairs and information flows will not be required. A modular approach for designing the INAP, therefore, will facilitate future protocol enhancements that will be needed when new SIBs are created or if existing SIBs need to be enhanced.

The relationship of the SIBs to the INAP now needs to be addressed. A modular "building block" approach is proposed for implementing the INAP, because:

– each SIB is modelled as a separate family of client/server pairs and information flows relating to them;

– not every network has to implement every SIB: if a SIB is not implemented in the network, then the corresponding client/server pairs and information flows will not be required;

– a modular approach would facilitate future enhancements (new SIBs or enhancements to existing SIBs).

Such a modular "building block" approach is readily available for the design of the IN Application Protocol – namely, the concept of the Application Service Element (ASE). This is defined in ISO 9545 as "a set of applications-functions that provides a capability for the interworking of application-entity-invocations for a specific purpose". In this case, the "specific purpose" is SIB-related communication. Thus, the elements of the protocol needed for each SIB could be defined by a set of one or more ASEs.

The paradigm that best seems to model the information flows between a client and a server is that of a request/reply interaction. The client requests some action to be performed by the server, which responds with the results when it has completed the task or informs the client that it cannot complete the task. Each client-server information flow will, of course, have its own specific requests and replies. If one were to define a protocol for every application that decided to model its communications using the request/reply paradigm, then there would be an unwieldy – potentially infinite – number of protocols to define and manage. Recognizing this, CCITT has standardized a mechanism – called the Remote Operations Service Element (ROSE) – where a separation is made between:

a) the common generic "vehicle" for the conveyance of a request (called, in technical jargon, the invocation of an operation) and the return of the response; and

b) the application-specific syntax and semantics of the requests and responses.

Item a) is the ROSE protocol specified in Recommendation X.229. Item b) is called the ROSE-User ASE and the guidelines for specifying these ASEs is, among other things, provided in Recommendation X.219. The next subclause will provide greater detail on the use of ROSE by INAP.

The previous subclause discussed how a Translation SIB might be modelled as (in general, a family) of client-server pair(s) – one located in the SCF and one in the SDF – correlated by information flow(s). Assuming that the SCF and SDF (and hence, the client and server) are implemented in different Physical Entities [e.g. in a Service Control Point (SCP) and a specialized data base respectively], then the appropriate client/server information flows need to be implemented as protocol. The protocol to realize these information flows could be defined by one (or more) ASE(s).

11.1.4 INAP Protocol Platform

In the near term, such a protocol should make use of existing protocol standards and platforms (e.g. TCAP, Q.932, X.217/219) wherever possible. However, to ensure a smooth evolution towards a common longer-term OSI-aligned platform, it should ideally be independent of any specific underlying communications protocols.

The "existing standards" (including TCAP and Q.932) are all based upon ISO 9545 "Application Layer Structure" and Recommendations X.219/229 Remote Operations (ROSE). The information flows between the client/server pairs are realized in INAP through the use of remote operations. The client requests the server to perform a procedure by encapsulating the specifics of the request in an INVOKE PDU. The results (if any) are returned in a RETURN RESULT or RETURN ERROR PDU, depending, respectively, on whether the specific request (i.e. operation invoked) could or could not be completed. Syntax errors in the INAP or ROSE protocol are signalled through the REJECT PDU.

The specific semantics of particular client/server procedures are defined through the use of the OPERATION and ERROR macros which are a notation for showing the relationship between the "user data" of various ROSE PDUs required to complete a remote procedure. The OPERATION MACRO describes for a given operation (referenced by an INTEGER or OBJECT IDENTIFIER value) the arguments (i.e. parameters) that accompany the operation invocation, the arguments (parameters) that are returned upon successful completion and the error causes that signal the inability to

complete an operation. Should an operation require a "linked operation" to complete, that is also denoted here. The arguments that further qualify the report of an error are specified through an ERROR MACRO. Details on OPERATION and ERROR macros and how to use them are provided in Recommendation Q.775 (the TCAP User's Guide) as well as in Recommendations X.219 and X.229.

Thus the client/server information flow(s) are specified in protocol by ASN.1 modules that specify, using the OPERATION and ERROR MACRO notations, a set of operations and errors that comprise the SIB capability.

11.2 Structure in the Application Layer

The communications needs for the IN applications are provided by an Application Entity (AE). This AE consists of several building blocks, one of which is TCAP whose purpose is to provide a mechanism to convey remote operations and their results. [In other environments, such as the use of the full OSI stack, there will be a need to have the Association Control Service Element (ACSE).] Each AE will also contain one (or more) TCAP-User ASEs which are modules of specifications which contain the definitions of the remote operations that make up the abstract syntax of the data exchanged.

Given these building blocks, there are several ways in which the AE can be structured. These are outlined below and the usefulness or deficiencies of each are pointed out.

11.2.1 The Monolithic Approach

In this arrangement, the entire applications protocol which realizes the information flows between all possible FEAs is defined as one large TCAP-User ASE. The requirement is that all operation and error codes have to be assigned unique local INTEGER values or globally unambiguous OBJECT IDENTIFIER values.

While this has the apparent advantage of simplicity, the disadvantage is that the interface between two FEs is not clearly distinguished and every node is required to support the entire protocol even if it will not use all of it. The reason is that there is no way using the current protocol to signal which subset of the total protocol each node supports without an explicit handshake at the start of a transaction preceding the exchange of data. Such explicit handshakes have been considered detrimental to signalling efficiency. Thus, even though the SCF and SDF may participate in only a limited number of inter-SIB client/server relationships, the network entities realizing these FEs has to support the entire set of information flows (which maps to the one application protocol defined by the single AE type).

Another disadvantage is the inability to easily borrow some useful TCAP-User protocol from other specification. Let us say that a remote operations-based authentication protocol has been defined somewhere which is found to be particularly useful. Should the INAP wish to borrow this protocol, rather than define its own version, it must ensure that importing the new set of operations/errors into INAP does not cause code value clashes.

Of course, INAP could define globally unique values (of the type OBJECT IDENTIFIER) for operation and error codes. This requires several extra octets to encode.

11.2.2 A Modular Approach

In designing any complex application protocol, the designer may choose to structure the protocol such that the AE is comprised of several TCAP-User ASEs. For example, the communications aspects of charging can be grouped together and separated from the aspects that have to do with a number translation. Such a structure allows the charging function to be used in combination with some other capability (e.g. collect digits) should that combination be found useful in some service scenario.

In such a "top-down" modular approach, the safest way to ensure that constituent TCAP-User ASEs can coexist in the same AE is to ensure that the operations and error are defined by TYPE in some module (specification). Values (integers) are assigned when these definitions are imported into the module defining the combination of ASEs that form a particular AE. A more constraining way to ensure uniqueness of local operation/error codes is to assign a range of code values (integers) to each ASE that is defined. (Of course, not all values in that range need be used.) Again, there is always the possibility of providing globally unique values by assigning operation/error codes to the type OBJECT IDENTIFIER.

The main advantage of using the definition of operations/errors by TYPE and assigning local values only within the modules into which they are imported is that there is only one definition of the syntax and information content of a particular operation. If the operation needs to be changed, it is done only once; in the other approach every replicated definition needs to be changed. If the old definition cannot be easily modified (say by the addition of some new parameter) then a new operation is created – and only once – and its definition is available to anyone wishing to use it. This eases administration and leads to a more structured method of specification.

11.2.3 Use of Application Context

This subclause outlines a further OSI-based structuring capability that has been defined by CCITT SG XI and which shall be adopted by various signalling protocols that use Transaction Capabilities. This structuring capability – called the Application Context – will be used in conjunction with the modular structure of the Application Layer described in clause 4.

In CCITT TCAP, a new information element corresponding to the Application Context has been defined whose purpose is to identify the TC-User ASEs that will be used during a given transaction. As described in clauses 4 and 6, this is necessary when the capabilities of the Application Layer have been divided into TCAP plus many TCAP-User ASEs, each of the latter being concerned with the communications for some specific capability. The Application Context will be included in the TCAP Begin and Continue messages, and also the End message for short (two message) transactions. Without this extra piece of information in a Begin message, there is no other way to signal "up-front" which particular TCAP-User ASEs will be required on a given transaction. The only way to know how to "route" an operation to the correct process or capability is to use a unique operation code value as an index to the type of capability/procedure that is being requested. Or, if there are "generic" operations, it is necessary to "look deeper" into an operation for a special parameter; or, if parameters are also made "generic", it is necessary to look for a special parameter value to determine the requested capability.

In the network, an instance of communication between any pair of Physical Entities will potentially concern many SIBs, depending upon the Physical Entities involved and the service feature(s) being executed. It may also be necessary to include communication capabilities (ASEs) defined elsewhere, e.g. from mobility services or supplementary services. The Application Context selected must include the ASEs for all the client/server interactions anticipated, plus, if so requested, those for mobility and supplementary services. In situations where a particular node supports only a certain specific set of capabilities, i.e. in situations where each FE is implemented in a separate physical entity, a different protocol (i.e. application context) is needed across the interfaces (information flows) between each FE pair. (However, if two FEs are at the same node, then their communications do not require a standardized communications protocol.) The AC signals, at the beginning of the transaction, which set of ASEs are potentially going to be used for protocol exchange between the two AEs. If there were "versions" of these ASEs defined, these could also be signalled at such a time.

With a monolithic protocol, all the capabilities are potentially required at every AE. There would be no flexibility in providing for just those functions (ASEs) that are found necessary at a given node.

Thus, the structuring principle being adopted by at least one SG XI application protocol is as follows:

- Operations and errors are defined by type in modules.
- Related operations and errors are imported by type into modules called ASEs.
- Sets of ASEs make up an AC. In the module where the AC is defined, non-overlapping values are assigned to the operation/error codes.
- The corresponding application protocol is defined as a set of possible Application Contexts.
- The AE at a physical node supports one, some or all of these ACs depending on the FEs allocated to it.
- Two AEs agree at the start of the transaction on the AC that will be used for that instance of communications.

11.3 Proposed Structure of the INAP

The IN Application Protocol should be structured in the modular way described above so that for any instance of communication (i.e. transaction) between two Physical Entities, an Application Context can be selected which contains those ASEs which are needed to complete the network capability requested of the SIBs. For example, between the SSF and SCF FEs the possible Application Context(s) would bind together, for a given transaction, the ASEs for charging, monitoring, basic call control, etc. Thus, the AC is a subset of the total IN protocol and specifies that portion needed for communicating between these two types of functional entities. In a similar way, one can specify the subset of the IN protocol needed between the SCF and the SRF FE types, and so forth.

The easiest modular structure is to group the procedures that constitute a client/server relationship for providing the capabilities of a SIB into one TC-User ASE. One can think of the "generic" description of the SIB (e.g. CONNECTION or USER INTERACTION, to use the example in subclause 2.1) as providing a "family" of operations with the individual procedures being the "specifiers".

Specifications are simplified if a single ASN.1 module contains the type descriptions of all the operation and errors using the MACRO notation. When specific ASEs are constructed for the AE at a physical entity, the operations and errors are given locally unique values. Combinations of ASEs are defined by an Application Context and the name of this context conveyed in protocol at the start of a transaction.

11.4 Protocol Assumptions

INAP will be a set of User ASEs of an X.219/229 Remote Operations (ROSE)-aligned Application Layer protocol. The ROSE functionality is provided in SS No. 7 by the Component Sub-Layer (CSL) of TCAP and in DSS 1 by the Facility IE of Recommendation Q.932. The Application Layer should be independent of the underlying transport mechanism (e.g. it is possible that TCAP over Recommendation X.25 could be used). The four TCAP Component Sub-Layer (CSL) Application Protocol Data Units (APDUs) which are fully aligned with ROSE will be used, namely:

- Invoke
- Return Result
- Return Error
- Reject

with each INVOKE operation being one of four possible classes of operations:

- Class 1: Success (Return Result) and failure (Return Error) of the Invoke operation is reported
- Class 2: Only success is reported
- Class 3: Only failure is reported
- Class 4: Neither success nor failure is reported.

It will be assumed that the TCAP Transaction Sub-Layer (TSL) will provide services to initiate (Begin), continue (Continue) and terminate (End for graceful completion, Abort for abnormal completion) transactions for a particular call and to package multiple ROSE PDUs together in a single message.

The TCAP-based Application Layer structure to be used for IN is depicted in Figure 16.

11.5 IN Application Part Structure

As discussed earlier, the INAP can be structured either as one monolithic ASE or as a set of several ASEs grouped by functionality. The discussion pointed out the inherent advantages of the latter approach. In short, dividing the operations into self-contained groupings of functions is useful if there is possibility of reuse of the groupings for a variety of different contexts depending on current, planned and potential long-term distribution of network capabilities.

This subclause discusses the criteria for determining the functional grouping of operations into ASEs.

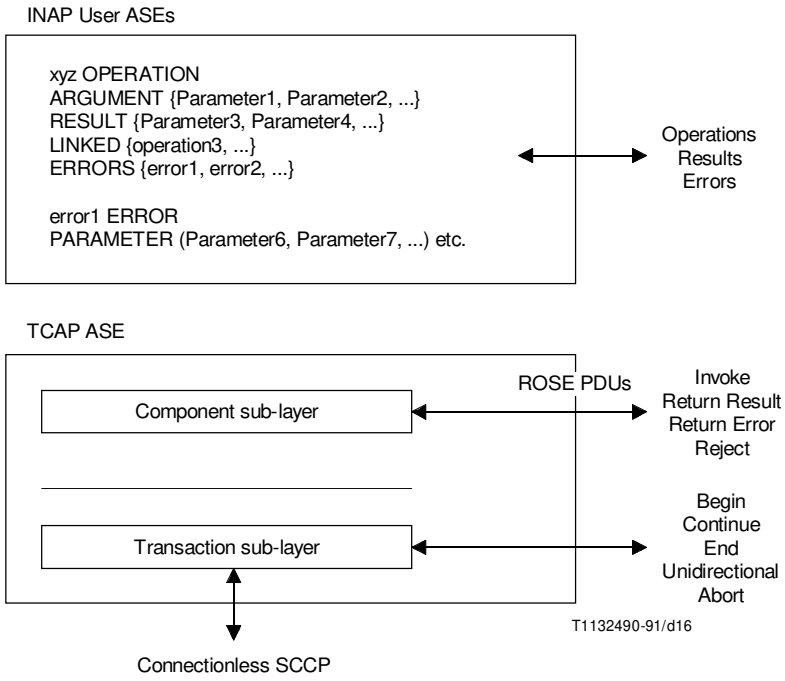


FIGURE 16/Q.1400

SS No. 7 Application Layer Structure

11.5.1 Criteria for Grouping Operations into ASEs

The following are guidelines which should be used to group operations into ASEs:

– *The relationship of the SIBs to the ASEs* – It may be possible that the SIB-ASE relationships are one-to-one, one to many or many to one. (Some SIBs, by their very nature, may be processed within a particular Functional Entity and therefore require no protocol.) However, a many-to-many relationship, in which operations related to SIBs are completely repackaged, does not seem practical. For example, packaging some charging operations with call processing operations and some charging operations with caller interaction operations may not lead to meaningful combinations and is not desirable.

– *Functional distribution* – One important aspect of IN is the ability to have flexible mapping of Functional Entities to Physical Entities, i.e. the ability to implement the Functional Entities co-located or separated by a standardized interface. Remote operations performed by the same pair of Functional Entities (FEs) have the potential of being grouped into one ASE, while those performed by different pairs of FEs should not be grouped together. For example, one might group operations needed to support the Announcement and Collect Info SIBs into one ASE, as both require an SCF-to-SRF interaction, but one would not want to group operations for the Translation (SCF-to-SDF interaction) and Basic Call Process (SCF-to-SSF) SIBs in the same ASE.

– *Modular reuse* – Reusability of ASEs in many different contexts is a guideline for good design. For example, a network traffic management ASE for traffic filtering can be useful both for an SCF-SSF and an SDF-SCF interface.

– *Objects* – Several other standards areas, most notably Network Management, have used object-oriented modelling techniques as a tool for deriving logically complete groupings of operations and for predicting potential future distribution of network functionality. Applying object-oriented analysis techniques, it appears useful to group operations acting on an end-to-end connection object in one ASE and to group operations acting on a path object of a connection in another ASE.

– *Future evolution* – Obviously, future capability sets will build upon the foundation which CS1 will provide. It must be possible to add new capabilities without having to alter the ASE definitions used for the previous Capability Sets. Changes to one ASE (e.g. adding new parameters to operations of the ASE or adding a new operation to the ASE) in a new version of the ASE should not impact the definitions of other ASEs. It will also be necessary for CS1 ASE development to take into account possible future distribution of network functionality. For example, it may seem expedient to combine call processing and caller interaction operations in the same ASE; however, such an ASE would preclude support of signalling connection between a Physical Entity (PE) containing a Service Control Function (SCF) and a stand-alone PE containing Specialized Resource Function (SRF) capabilities.

11.5.2 Criteria for Identification of Operations

The formation of INAP operations, or the remote operations questions to be asked across IN interfaces, also merit careful thought. Decisions must take into account current and potential physical configurations, protocol flexibility and real-time performance issues. For example, when determining whether call processing and call charging parameters should be combined within one operation, the following must be taken into account:

– Does charging need to be supported by a multiplicity of call processing and non-call processing (e.g. caller interaction) operations in other ASEs?

– What if future systems designers decide to physically separate charging functionality from call processing functionality?

– What if another TCAP-User or ROSE-User would like to "borrow" INAP's charging capabilities? How easy will modular reuse turn out to be?

– How will real-time performance be impacted?

It is a good practice for the operations to ask as specific a question as possible. Since the TCAP TSL allows multiple components to be packaged within one message, required modularity can be supported with no impact on the number of transactions required and with minimal overhead. Of course, this is assuming that such modularity can be shown to provide a future benefit and does not make the protocol unnecessarily complex.

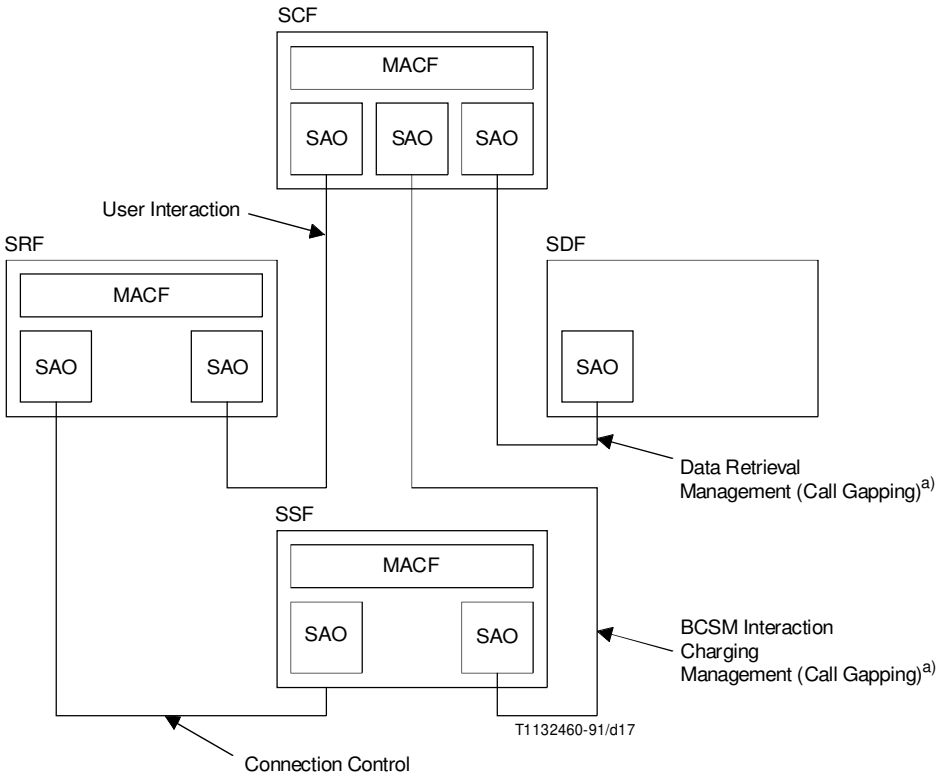
The following points must also be considered in determining the set of INAP operations:

– If an operation is so generic that a multiplicity of potential types of results are envisioned, then it is likely that more than one operation is required. On the other hand, if multiple operations are proposed which ask the same question and expect the same results, then only one operation should be required.

– In theory, operations should be specific enough so that the class of the operation is identifiable from the operation code. The operation code is used as an index into the MACRO notation definition of the operation. The presence/absence of the keywords RESULTS and ERRORS determines the class of the operation. Note that the class of the operation is not decided dynamically, but rather is accepted by both peers as part of the ASE definition.

11.6 Hypothetical Example

A hypothetical example to illustrate the use of several ASEs for INAP is shown in Figure 17. (None of the ASEs shown in Figure 17 should be treated as a specific recommendation for INAP.) Note that the protocol for management (call gapping) may be applicable in several relationships, and should therefore be specified as an ASE.



SCF Service Control Function
SRF Specialized Resource Function
SDF Service Data Function
SSF Service Switching Function
MACF Multiple Association Control Function
SAO Single Association Object
ASE Application Service Element
BCSM Basic Call State Model

a) Reused ASE.

FIGURE 17/Q.1400

Hypothetical Example to Illustrate Usage of ASEs

12 Compatibility Mechanisms and Rules in SS No. 7 and DSS 1

Subclauses 12.1 through 12.4 relate to non-OSI-based signalling and OA&M protocols. Subclause 12.5 addresses OSI (ROSE-based) protocols.

12.1 Background

The wide scope of the signalling system requires that the total system include a large diversity of functions and that further functions can be added to cater for extended future applications. As a consequence, only a subset of the total system may need to be used in individual application.

A major characteristic of the signalling system is that it is specified with a functional structure to ensure flexibility and modularity for diverse applications within one system concept. This allows the system to be realized as a number of functional modules which could ease adaptation of the functional content of operating SS No. 7 and/or DSS 1 to the requirements of particular applications.

The CCITT specifications of the signalling system specify functions and their use for international operation of the system. Many of those functions are also required in typical national applications. Furthermore, the system to some extent includes features that are particular to national applications. The CCITT specifications thus form an internationally standardized base for a wide range of national applications of common channel signalling.

SS No. 7 is one common channel signalling system and DSS 1 is one digital access signalling system. However, as a consequence of their modularity and their intended use as bases for national applications, the systems may be applied in many forms. In general, to define the use of a system in a given national application, a selection of the CCITT specified functions must be specified depending on the nature of the application.

SS No. 7 and DSS 1 are evolutionary signalling systems which have undergone a number of enhancements. To allow ease of evolution it has been necessary to incorporate a number of compatibility mechanisms in various functional elements, and to apply a number of compatibility rules to protocol enhancement. Detailed specifications of the compatibility mechanisms in each functional element are given in the appropriate Q.700-Series and Q.900-Series Recommendations. An overview is given in this Recommendation.

Compatibility rules which apply to all functional elements of SS No. 7 and DSS 1 are detailed in the following text.

12.2 Evolutionary Requirements

In application protocols (e.g. ISDN-UP, ASEs), the main evolutionary requirement is the ability to add new subscriber services, new administration and network services to the protocol.

In the SCCP and MTP, the evolutionary requirements are different in that initial versions provide basic transport functions which are generally stable. The main enhancements have been in the management aspects of the protocols.

Although the evolutionary requirements are different across the elements of SS No. 7, it is possible to incorporate certain common mechanisms in the various functional elements.

12.3 Forward and Backward Compatibility

Compatibility mechanisms can be considered as being either:

- Forward compatibility mechanisms; or
- Backward compatibility rules.

Forward compatibility mechanisms are defined as a scheme to enable a version of a protocol to communicate effectively with and interwork with future versions of the protocol. That is, a version of a protocol should not restrict future protocols from providing extra capabilities.

Backward compatibility rules are defined as a scheme to ensure that future versions of the protocol will be able to send protocol messages to the previous version which will be understood and fully processed by the node supporting the previous version. That is, future versions of a protocol must allow earlier versions to operate with it and not reduce the earlier version's service level.

12.4 Compatibility Rules for SS No. 7 and DSS 1

The following compatibility rules are applied to each element of SS No. 7 (e.g. ISDN-UP) and DSS 1 when protocols are enhanced, or when a subsequent version of a protocol is prepared.

12.4.1 Rules for Forward Compatibility

All future versions of CCITT Recommendations for non-OSI-based SS No. 7 protocol elements (e.g. SCCP, ISDN-UP, etc.) and DSS 1, from 1992 onwards, shall include a mechanism for forward compatibility. The following list contains the basic requirements of the mechanism:

i) include the ability to send a message indicating that the received information was not understood in response to an unrecognized message or parameter;

ii) send this message to the node responsible for the confusing information if the necessary routing information is available;

iii) for existing protocols, state the action to be taken on receipt of spare or reserved values of defined parameters, e.g. treat as appropriate default values, transmit them unchanged at intermediate nodes, or ignore them at end nodes;

iv) when defining new messages, parameters or parameter values to support a new function, include in the specification the action to be taken when a confusion message is received in response to the new message, parameter or value indicating that the information was not understood;

v) only send messages requiring an acknowledgement a limited number of times (e.g. three). If no response is received, the sending signalling point should assume that the facility is not available and inform local management;

vi) state that all new messages shall have the ability to add new optional fields; and

vii) unallocated codes of defined fields should be examined and handled as spare codes.

Note that the 1992 ISUP Recommendations (see the Q.76x–Series Recommendations) contain a special compatibility procedure. It uses an instruction indicator, which includes information about the handling of a parameter or message that is not recognized (e.g. discard, pass-on, send Confusion). It is sent with every new message or parameter. For parameters containing new values, it is assumed that the instruction indicator for the whole parameter can be used for all values within the parameter. For existing messages, parameters and parameter values, the required action if unrecognized information is received is given in tabular form.

12.4.2 Rules for Backward Compatibility

All future versions of CCITT Recommendations for non-OSI-based SS No. 7 elements (e.g. SCCP, ISDN-UP, etc.) and DSS 1, from 1992 onwards, shall include a mechanism for backward compatibility. The following list contains the basic requirements of the mechanism.

12.4.2.1 Existing Messages

i) The ability to receive all existing messages shall be possible, since the removal of a message implies the removal of a function.

ii) The effect of receiving any existing message, parameter or function, in a new version, must be the same as that in previous versions. The effects of new parameters or parameter values will thus be purely additive.

12.4.2.2 Existing Message Parameters

Message parameters consist of three basic types and occur in the pre-defined order: mandatory fixed length, mandatory variable length and optional fixed or variable length.

The following rules shall apply:

i) no fixed length parameter of the mandatory type shall be changed to variable length;

ii) no variable length parameter of the mandatory type shall be changed to fixed type;

iii) optional parameters shall not become mandatory;

iv) mandatory parameters shall not become optional;

v) additional mandatory parameters shall not be added to a message;

vi) in existing messages whose format allows optional parameters, additional optional parameters are allowed;

vii) existing mandatory parameters shall not be removed from existing messages;

viii) the range of any parameter for an existing message shall not be reduced;

ix) the meaning of any defined parameter value shall not be changed on an existing message; and

x) there are no restrictions on the parameters for new messages.

12.4.2.3 New Messages

i) New messages may be added after a Recommendation has been published, however, nodes that do not recognize these new messages will respond with a message indicating that the information was not recognized.

ii) The "information not recognized" message shall never be sent in response to a received "information not recognized" message, nor in response to other recognized messages received in the wrong call state.

Appropriate default action shall be defined to handle these situations.

12.4.2.4 New Parameters

New optional parameters can be added to existing messages after a Recommendation has been published, however, nodes that do not recognize these new parameters will respond with a message indicating that the information was not recognized.

12.4.2.5 New Parameter Fields

New fields may be added to, or spare fields used in existing parameters after a Recommendation has been published, however, nodes that do not recognize these new fields will respond with a message indicating that the information was not recognized.

12.4.2.6 New Parameter Values

Previously spare, reserved or unallocated parameter values can be used after a Recommendation has been published; these will be treated as defined in 12.4.1 iii).

12.4.3 Handling of Unrecognized Information

When a new protocol, message or information element is created, a rule is required on a per message and information element basis to define the action on receipt of unrecognized information. This rule needs to be applied to unrecognized messages, unrecognized information elements within messages, and unrecognized values within recognized information elements.

The actions defined for receipt of an unrecognized message/information element could be:

– discard message/information element;

– discard/ignore information element within a recognized message;

– default to a known general value (e.g. on receipt of an ISDN-UP IAM with an unrecognized calling party category, the value could be defaulted to "Unknown");

– send a "Confusion" message;

– terminate the call/transaction; or

– inform management.

12.4.4 Increase in the Length of Optional Parameters

If a parameter is used as an optional parameter in all messages that it appears, the length of the parameter can be increased. The older version of the protocol would be able to function as it does today, assuming it ignores the extra bits or a suitable extension method has been defined. The newer version would have to check the length of the parameter to determine if the added information was present.

Protocols which use coding rules which are based on ASN.1 (e.g. TC) are not subject to this rule.

12.4.5 Processing of SS No. 7 Messages with Unrecognized SIO Information

To enable signalling points implemented to the 1988 Recommendations to interwork with signalling points implemented to earlier Recommendations when a message containing an unrecognized service information octet (see 14.2/Q.704) is received, the message is discarded.

12.4.6 Unacknowledged Messages

Where a function requires an acknowledgement to a message in order to continue, if no response is received the function sends the message for only a limited number of times. The sending signalling point should assume that the function is not available, and inform local management.

12.4.7 Processing of Spare Fields

For those functions which define fields or sub-fields in signalling messages as spare or reserved, the following rules for processing of these fields apply.

At a node generating a signalling message, all spare and reserved fields are set to zero. At transit nodes, spare or reserved fields may be passed on transparently. At the destination node, the spare and reserved fields are not examined.

12.5 Application Protocol Enhancement Mechanism (ROSE-based protocols)

It is envisaged that minor extensions to an application protocol may be needed from time to time. An abstract syntax is extended if its associated type is extended (i.e. if a choice type, it can be extended by adding a new component or extending an existing one). One way of extending a PDU (or any structure type) is to extend the type of any of its components. In supporting such extensions, care needs to be taken to ensure that the extensions are indeed minor. Therefore, the following types of extensions to the Abstract Syntax might be considered minor:

- addition of an information element which may enhance an activity but is not essential to performing the basic activity (e.g. list of additional routing options); or
- addition of an information element to add a capability which is not essential to the base capability (e.g. addition of “Name” in addition to “Number” for terminal display purposes).

In the above cases, a new Application Context name need not be defined, however forward compatibility procedures for dealing with the unknown information must exist at the receiving application process.

The following types of extensions might be considered major:

- addition of a new procedure; or
- fundamental alteration of a procedure (e.g. “do this procedure twice”).

In these cases, a new Application Context name should be defined.

When the changes are judged by the application designer not to warrant a new Application Context name, then an Extension Field containing a “criticality factor” may be used to extend the syntax of an existing PDU. This requires forward compatibility procedures to be available at the receiving end.

The use of a “criticality factor” enables the receiving entity to determine appropriate reaction to the extension received should it not be understood.

An example of a generic extension mechanism is described by the following ASN.1 specification:

```
Extension-Mechanism-Example-1
        { ccitt recommendation q 1400 modules(0) extension-example(1)
          version1(0) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
        ExtensionField ::= SEQUENCE { type EXTENSION,
        value [1] ANY DEFINED BY type }
        EXTENSION MACRO  ::=
        TYPE NOTATION      ::= ExtensionType Criticality
        VALUE NOTATION     ::= value (VALUE CHOICE {
                                    private-extension INTEGER,
                                    standard-extension OBJECT IDENTIFIER })
        ExtensionType      ::= "EXTENSION-SYNTAX" type | empty
        Criticality        ::= "CRITICALITY" value (CriticalityType)
        CriticalityType    ::= ENUMERATED { ignore(0),
                                            abort(1) }
END
```

Any application protocol wishing to use this extension should IMPORT the ExtensionField from the above described module into the module where such extensions are needed.

The following example illustrates the use of this mechanism for ROSE operations. The ASN.1 has been substantially simplified in order to help focus on the extension mechanism. The same approach applies to errors.

Original operation which cannot be extended:

```
nameOfOperation OPERATION
ARGUMENT SEQUENCE { x1, x2, x3 }
RESULT SEQUENCE { y1, y2 }
ERRORS etc.
```

Revised operation which can be extended:

```
nameOfOperation OPERATION
ARGUMENT SEQUENCE { X1, X2, X3, SET OF { ExtensionField } OPTIONAL }
RESULT sEQUENCE { Y1, Y2, SET OF { ExtensionField } OPTIONAL }
ERRORS etc.
```

It should be recognized that there is ongoing work in this area in ISO, and that other alternatives may emerge. However, the above mechanism is considered suitable for use in signalling and OA&M application protocols using ROSE.

13 References

– Recommendation X.200 *Reference Model of Open Systems Interconnection for CCITT Applications.*

  ISO/IEC 7498-1 *Information Processing Systems – Open Systems Interconnection – Basic Reference Model.*

– Recommendation X.217 | ISO/IEC 8649 *Information Technology – Open Systems Interconnection – Service Definition for the Association Control Service Element.*

– Recommendation X.227 | ISO/IEC 8650 *Information Technology – Open Systems Interconnection – Protocol Specification for the Association Control Service Element.*

  ISO/IEC 10035 *Information Technology – Open Systems Interconnection – Connectionless ACSE Protocol Specification (A-UNITDATA protocol).*

– Recommendation X.219. *Remote Operations: Model, Notation and Service Definition.*

  ISO/IEC 9072-1 *Information Processing Systems – Text Communication – Remote Operations – Part 1: Model, Notation and Service Definition.*

– Recommendation X.229 *Remote Operations: Protocol Specification.*

  ISO/IEC 9072-2 *Information Processing Systems – Text Communication – Remote Operations – Part 2: Protocol Specification.*

  ISO/IEC 9545 *Information Processing Systems – Open Systems Interconnection – Application Layer Structure.*

– Recommendation X.216 *Presentation Service Definition for Open Systems Interconnection for CCITT Applications.*

  ISO/IEC 8822 *Information Processing Systems – Open Systems Interconnection – Connection-Oriented Presentation Service Definition Amendment 1: Connectionless Mode Presentation Service.*

– Recommendation X.226 *Presentation Protocol Specification for Open Systems Interconnection for CCITT Applications.*

  ISO/IEC 8823 *Information Processing Systems – Open Systems Interconnection – Connection-Oriented Presentation Protocol Specification.*

  ISO/IEC 9576 *Information Technology – Open Systems Interconnection – Presentation Connectionless Protocol to Provide the Connectionless Presentation Service.*

– Recommendation X.690 (to be published), *Information Technology – Open Systems Interconnection – Specification of ASN.1 Encoding Rules: Basic Encoding Rules.*

  ISO/IEC 8824 *Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).*

– Recommendation X.209 *Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*

  ISO/IEC 8825 *Information Technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*

– Recommendation X.650 *Open Systems Interconnection (OSI) – Reference Model for Naming and Addressing.*

  ISO/IEC 7498-3 *Information Processing Systems: Open Systems Interconnection – Basic Reference Model – Part 3: Naming and Addressing.*

– Recommendation X.700 *OSI Management Framework.*

– Recommendation Q.700 *Introduction to CCITT Signalling System No. 7.*

– Recommendations Q.701-Q.706 *CCITT Recommendations for the Message Transfer Part (MTP) of Signalling System No. 7.*

– Recommendation Q.711-Q.716 *CCITT Recommendations for the Signalling Connection Control Part (SCCP) of Signalling System No. 7.*

– Recommendation Q.750-Series, *Operations, Maintenance and Administration Part (OMAP) of Signalling System No. 7.*

– Recommendations Q.761-Q.764, Q.766 *CCITT Recommendations for the ISDN User Part (ISDN-UP) of Signalling System No. 7.*

– Recommendations Q.771-Q.775 *CCITT Recommendations for the Transaction Capabilities Application Part (TCAP) of Signalling System No. 7.*

– Recommendation Q.931 *ISDN User-Network Interface Layer 3 Specification for Basic Call Control.*

– Recommendation Q.932 *Generic Procedures for the Control of ISDN Supplementary Services.*

– Recommendations Q.81x and Q.82x *CCITT Recommendations, Network Management Service.*

  ISO DIS 10026 *Information Technology OSI – Distributed Transaction Processing – Part 1: OSI TP Model.*

  Information Technology OSI – Distributed Transaction Processing – Part 2: OSI TP Service.

  Information Technology OSI – Distributed Transaction Processing – Part 3: Protocol Specification.

14 List of Acronyms

For the purpose of this Recommendation, the following abbreviations are used:

AARE	Application Association Response
AARQ	Application Association Request
AC	Application Context
ACPM	Association Control Protocol Machine
ACSE	Association Control Service Element
AE	Application Entity
AEI	Application Entity Invocation
AFI	Authority and Format Identifier
ALS	Application Layer Structure
AMI	Application Management Interface
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Process Invocation
AS	Abstract Syntax
ASE	Application Service Element
ASN.1	Abstract Syntax Notation One
AUDT	Application Unitdata
BCSM	Basic Call State Model
BER	Basic Encoding Rules
CEI	Connection Endpoint Identifier
CL	Connectionless
CL-NS	Connectionless Network Service
CL-SCCP	Connectionless SCCP
CMIP	Common Management Information Protocol
CO	Connection-Oriented
CO-NS	Connection-Oriented Network Service
CO-SCCP	Connection-Oriented SCCP
CRN	Call Reference Number
CSL	Component Sub-Layer
DCS	Defined Context Set
DFP	Distributed Functional Plane
DSAP	Data Link Service Access Point
DSP	Domain Specific Part
DSS 1	Digital Subscriber Signalling 1
DTMF	Dual-Tone Multi-Frequency

FE	Functional Entity
FEA	Functional Entity Action
FTAM	File Transfer and Access Management
GFP	Global Functional Plane
GT	Global Title
HLR	Home Location Register
IAM	Initial Address Message
IDI	Initial Domain Identifier
IDP	Initial Domain Part
IE	Information Element
IN	Intelligent Network
INAP	Intelligent Network Application Part
ISDN	Integrated Services Digital Network
ISDN-UP	ISDN User Part
ISP	Intermediate Service Part
ISUP	Integrated Services User Part
LAPD	Link Access Protocol – D channel
LME	Level Management Entity
LMI	Level Management Interface
MACF	Multiple Association Control Function
MAP	Mobile Application Part
MHS	Message Handling System
MIB	Management Information Base
MMI	Man-Machine Interface
MSC	Mobile Switching Centre
MT	MTP Tester
MTP	Message Transfer Part
NC	Network Connection
NS	Network Service
NSAP	Network Service Access Point
NSP	Network Service Part
OM	Operation and Maintenance
OMAP	Operations, Maintenance and Administration Part
OMASE	Operations, Maintenance and Administration Service Element
OSI	Open Systems Interconnection

PABX	Private Automatic Branch Exchange
PAI	Protocol Addressing Information
PC	Point Code
PC	Presentation Context
PDU	Protocol Data Unit
PDV	Presentation Data Value
PhSAP	Physical Service Access Point
PPDU	Presentation Protocol Data Unit
PSAP	Presentation Service Access Point
QOS	Quality of Service
ROSE	Remote Operations Service Element
RTSE	Reliable Transfer Service Element
SA/NC	Signalling Area Network Code
SACF	Single Association Control Function
SAO	Single Association Object
SAP	Service Access Point
SAPI	Service Access Point Identifier
SCCP	Signalling Connection Control Part
SCF	Service Control Function
SCP	Service Control Point
SDF	Service Data Function
SDL	Specification and Description Language
SI	Service Indicator
SIB	Service Independent Building-block
SIO	Service Information Octet
SLP	Service Logic Program
SMAE	Systems Management Application Entity
SMAP	Systems Management Application Process
SMSI	Systems Management Service Interface
SRF	Specialized Resource Function
SS	Signalling System
SSAP	Session Service Access Point
SSF	Service Switching Function
SSN	Sub-System Number
ST	SCCP Tester

TC	Transaction Capabilities
TC	Transport Connection
TCAP	Transaction Capabilities Application Part
TD-PPDU	Transfer Data – Presentation Protocol Data Unit
TEI	Terminal Endpoint Identifier
TMN	Telecommunications Management Network
TP	Transaction Processing
TP	Transport Protocol
TS	Transfer Syntax
TSAP	Transport Service Access Point
TSL	Transaction Sub-Layer
TUP	Telephone User Part
VLR	Visited Location Register