ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-86

GENERIC DATA PRESENTATION

SERVICES DESCRIPTION AND PROTOCOL DEFINITION

March 1983

Free copies of this document are available from ECMA,
European Computer Manufacturers Association
114 Rue du Rhône – 1204 Geneva (Switzerland)

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-86

GENERIC DATA PRESENTATION

SERVICES DESCRIPTION
AND
PROTOCOL DEFINITION

March 1983

BRIEF HISTORY

This Standard has been developed by TC23 and adopted as an ECMA Standard at the General Assembly of Dec. 16, 1982.

This Standard ECMA-86 constitutes the present ECMA position as a technical contribution to ISO and CCITT to achieve world-wide standardization. However, ECMA does not yet consider this document to be a basis for product compliance as there is still international work going on.

Introduction

INTRODUCTION

This Standard ECMA-86 is one of a series of standards for Open Systems Interconnection.

Open Systems Interconnection standards are intended to facilitate homogeneous interconnection between heterogeneous information processing systems.

The Standard is within the framework for the coordination of standards for Open Systems Interconnection which is defined by ISO 7498.

It is based on the practical experience of ECMA member companies world-wide, and on the results of their active participation in the current work of ISO, the CCITT, and national standard bodies in Europe and the USA. It represents a pragmatic and widely based consensus.

A particular emphasis of this Standard is to specify the homogeneous externally visible and verifiable characteristics needed for interconnection compatibility, while avoiding unnecessary constraints upon and changes to the heterogeneous internal design and implementation of the information processing systems to be interconnected.

In the interests of rapid and effective standardization, the standard is orientated towards urgent and well understood needs. It is intended to be capable of modular extension to cover future developments in technology and needs.

TABLE OF CONTENTS

Page

GENERAL

1. GENERAL 1

1.1 Scope 1
1.2 Field of Application 1
1.3 References 1

SERVICE

2. SERVICE 2

2.1 Service Overview 2

2.1.1 Presentation-connection Establishment Facility 4
2.1.2 Presentation-connection Termination Facility 4
2.1.3 Enclosure Control Facility 4
2.1.4 Negotiation Facility 5
2.1.5 Information Handling Facilities 5
2.1.6 Dialogue Control Facility 5
2.1.7 Summary Table of Service Primitives for GPS 5
2.1.8 Subsets 7

2.2 Service Description 8

2.2.1 Functional Phases of the Generic Presentation Service 8
2.2.2 Establishment of Presentation-Connection 14
2.2.3 Termination of Presentation-Connection 19
2.2.4 Enclosure Control Facility 21
2.2.5 Negotiation Facility 25
2.2.6 Single-interaction Negotiation 29
2.2.7 Multiple-interaction Negotiation and Negotiation Enclosure 32
2.2.8 Information Handling Facilities and Transfer Enclosure 39
2.2.9 Service Exception Conditions 43
2.2.10 Diagnostic Information 43

PROTOCOL

3. PROTOCOL 46

3.1 Protocol Overview 46

3.1.1 Protocol Messages 46
3.1.2 Summary List of Protocol Messages 47
3.1.3 Conventions and Notation for Definition of Protocol Messages 48
3.1.4 Mapping of Messages into Lower Layer Service 48

3.2 Protocol Message Definitions 48

Table of Contents (cont'd)

Page

3.2.1 CONNECT Protocol Messages 48
3.2.2 RELEASE Protocol Messages 50
3.2.3 DISCONNECT Protocol Message 52
3.2.4 CHANGE ENCLOSURE Protocol Messages 52
3.2.5 CLOSE/ABORT-ENCLOSURE Protocol Structure 53
3.2.6 PERFORM NEGOTIATION Protocol Messages 54
3.2.7 NEGOTIATION ENCLOSURE Protocol Messages 55
3.2.8 Information Handling Facilities 57

3.3 Protocol Encoding 57

3.3.1 Protocol Message Structure 57
3.3.2 Presentation Header 58
3.3.3 Presentation Message Content 58
3.3.4 Parameter Encoding 58
3.3.5 Message Type Codes 62
3.3.6 Parameter Encoding Details 62
3.3.7 Special Parameter Encodings 65
3.3.8 Values of Severity in Diagnostic Parameters 67
3.3.9 Values of Reason in Diagnostic Parameters 67

3.4 Session Service Mapping 68

3.4.1 Summary of Usage of Session Services 68
3.4.2 Connection Mapping 68
3.4.3 Session Connection Establishment 68
3.4.4 Session Connection Termination 69
3.4.5 Other Session Service Facilities 70

APPENDICES

APPENDIX A - BRIEF DESCRIPTION OF THE REFERENCE MODEL FOR OPEN SYSTEMS INTERCONNECTION 71

APPENDIX B - INDEX AND GLOSSARY OF TERMS 76

APPENDIX C - SERVICE DESCRIPTION TECHNIQUE 81

APPENDIX D - ADDITIONS TO STANDARD ECMA-75 SESSION PROTOCOL 83

1 General

1. GENERAL

1.1 Scope

This Standard ECMA-86:

a) defines the terminology, concepts, descriptive model and notation for the Generic Services in the Presentation Layer of Open Systems Interconnection,

b) defines in abstract form the interactions between two presentation-service-users (p-users) via the Generic Presentation Services of Open Systems Interconnection, in Section 2, including the relationship to more specific facilities such as those applicable to Virtual Terminal Service or Data Presentation Service.

c) defines the protocol between two presentation entities (p-entities) providing the generic presentation services (p-service), in Section III, including the usage of services of Session Layer of OSI in support of the generic presentation services.

This Standard ECMA-86 does not define any other interactions between a presentation service user and the presentation service. Related ECMA Standards will define how the specialized requirements of particular presentation services are related to the Generic Presentation Services.

This Standard ECMA-86 is not an implementation specification for information processing systems.

1.2 Field of Application

This Standard ECMA-86 is provided in order to be referenced by other ECMA Standards specifying presentation protocol(s) for specialized presentation services.

1.3 References

ECMA-6 : 7-Bit Input/Output Coded Character Set

ECMA-75: Session Protocol

ISO 7498 Reference Model of Open Systems Interconnection

2 Service

2. SERVICE

2.1 Service Overview

The Presentation Layer Service provided within the Scope of this Standard is grouped into a number of Facilities as described in the sub-clauses below. A table of the Service Primitives which support these Facilities is given in 2.1.7.

The technique used for the description of the service primitives is given in Appendix C.

The following text and diagram give a general model of the Presentation Layer and of the services intended to assist the understanding of the later detailed description. For further information on the general OSI Model, refer to Appendix A.

The Generic Presentation Service (GPS) is a set of common service facilities of the Presentation Layer of the ISO OSI Model. The GPS is dependent on the underlying Session Services to establish and maintain communication between the presentation service users (p-users). Once a presentation-connection has been established between these p-users, all communication between them on this connection takes place according to the rules of the GPS and other service parameters selected and agreed by them. The p-service is accessed via an interface (which is generally implementation-dependent and as such may be different at each end of the connection) which functionally reflects (at least) the features of the service facilities being used. Figure 1 illustrates the p-layer and the concepts outlined above.

The range of capabilities within the generic definition of the GPS which can be chosen by the p-users is very large and the required set is established through negotiation; to assist this process there are major defined groupings known as specific presentation services (an example is "Virtual Terminal Service") and possibly further sub-groupings known as Classes of Service within these. Some of these service parameters may be modifiable by re-negotiation during the duration of a presentation-connection. These is also a subset concept for the GPS facilities themselves.

The remainder of Section II describes the generic features of the presentation services offered to the presentation service users. The description is done in terms of elementary operations, known as Service Elements, on a conceptual service interface. The concept of such an interface is convenient for describing the functional features of the GPS but this Standard gives no conformance rules for it. The service elements described in this clause are available to all specific presentation services although there are some features which will not be used at all by some. A Service Element consists, in general, of a sequence of service primitives on the conceptual service interface, a service primitive, usually abbreviated to just primitive, being an indivisible event on one service access point.

The method defined in Appendix C is used for the description of the service elements and primitives, etc.; this does not, of course, impose any requirement on an implementation of a real service interface to adopt this convention or these formats.

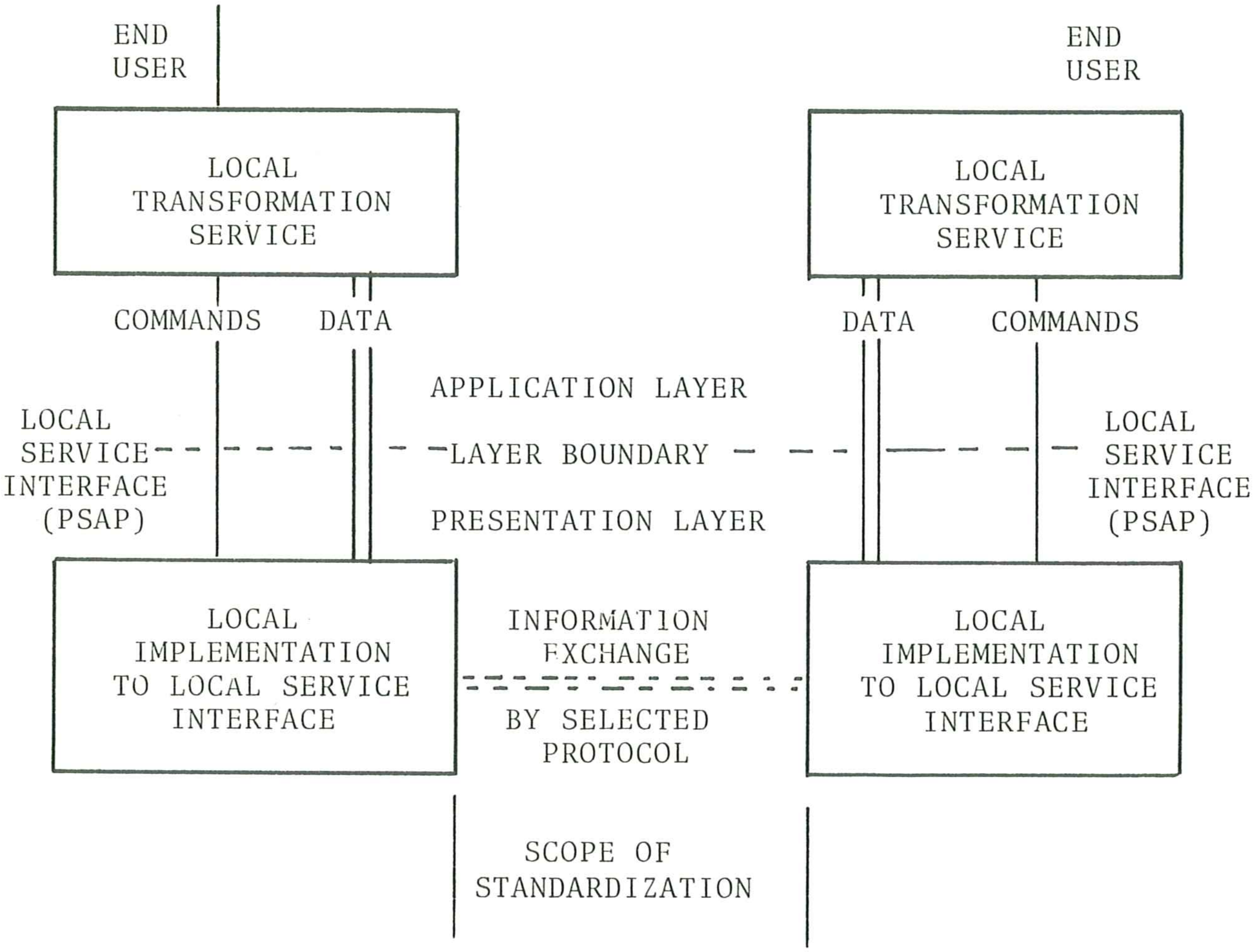

END USER
END USER
LOCAL TRANSFORMATION SERVICE
LOCAL TRANSFORMATION SERVICE
COMMANDS
DATA
DATA
COMMANDS
APPLICATION LAYER
LOCAL SERVICE INTERFACE (PSAP)
LAYER BOUNDARY
LOCAL SERVICE INTERFACE (PSAP)
PRESENTATION LAYER
LOCAL IMPLEMENTATION TO LOCAL SERVICE INTERFACE
INFORMATION EXCHANGE
BY SELECTED PROTOCOL
LOCAL IMPLEMENTATION TO LOCAL SERVICE INTERFACE
SCOPE OF STANDARDIZATION

Figure 1 Scope of Presentation Layer Standardization

Explanation of Figure 1

This diagram is a schematic (obviously very simplified but adequate for the present purpose) of the exchange of information between two End Users, human or system, by some Presentation Layer OSI Standard. The only item in this diagram which is being formally standardized is the protocol involved in the information exchange between the presentation entities, i.e. the line in the bottom centre of the picture.

An implementation will provide a local service interface to users of a Presentation Layer service. These users (which will in general not be the end users) in turn are free to provide (above Presentation Layer) a local transformation service which may then be used by other applications, used for generating displays on real devices, etc. The Standard is concerned only with the information exchange at bottom centre and in no way is it concerned with the local service interface or with any local transformation service.

However, in order to define the Standard in understandable terms it is desirable (if not necessary) to introduce a conceptual model and a set of actions affecting the information contained in this conceptual model. The description of the functional capabilities for which the protocol is designed to cater, in terms of actions on this model, is in itself a definition of a conceptual service interface. However compliance with the Standard in no way requires the local implementation to provide a particular representation of this conceptual service interface - it is free to choose its own syntax and its own "level" of interface, which may well be a macro level far above the basic level used in the description of the Standard. Furthermore, the interpretation of the data (as opposed to control functions) carried by the Presentation Layer protocol is not defined by the Standard so that local transformation services may be added at both ends and primitive protocols used to support very sophisticated modes of usage, e.g. sophisticated devices, without violating the Standard.

2.1.1 Presentation-connection Establishment Facility

Enables establishment of a p-connection with selection or negotiation of important characteristics of the p-service to be provided, including designation of a specific presentation service/protocol to operate within the framework of the Generic Presentation Protocol (GPP) and major options of the GPP itself.

2.1.2 Presentation-connection Termination Facility

Enables termination of the p-connection. Termination can be Orderly, possibly negotiated between the two p-users, or Forced, possibly destructive, by unilateral action of one p-user. The p-connection can also be lost due to exception condition in the Presentation-Layer or in the lower layer supporting the Presentation-Layer (Session-Layer).

2.1.3 Enclosure Control Facility

Usually the activity performed on a p-connection will be divided in time into a succession of contained phases known as enclosures; interaction activities from one enclosure are prevented from "penetrating" into the following one by a controlled transition between enclosure. This does not prevent the passing of, for example, parameters of the p-connection, from one enclosure to another.

One particular form of enclosure, the transfer-enclosure, is a feature of any p-connection of GPP since it is in transfer-enclosure that "useful" information handling activities are performed, using an agreed set of presentation parameters known as a presentation environment(PE).

As well as the controlled and agreed transition between enclosures, means are provided for a unilateral abort of an enclosure by one p-user.

2.1.4 Negotiation Facility

Two levels of Negotiation Facility are included in this Standard; their use is subject to agreement at the time of establishment of the p-connection (and may itself be subject to re-negotiation later). The simplest level provides only a "single-interaction" negotiation; the higher level provides in addition a "multiple-interaction" negotiation performed within a negotiation-enclosure, enabling a complex negotiation procedure to be performed where this is required.

2.1.5 Information Handling Facilities

The nature of information handling is not standardized by this Standard but it provides the means of controlling the context within which such activity can occur. Two levels are provided, the simplest provides for only a single presentation environment (PE) for use in the transfer-enclosure (although the parameters of this PE can be changed if negotiation is available). The advanced level, multiple-pe option, enables a choice of PE from a set of PEs to be made when the transfer-enclosure is entered.

2.1.6 Dialogue Control Facility

The requirements for dialogue control for a particular presentation service operating within the framework of the Generic Presentation Protocol are specific to the nature of that service (and hence of the operations performed by the associated protocol). Examples of the facilities are token management, synchronization, and re-synchronization. Each specific service and protocol is free to determine, in conjunction with the p-service-users (a-entities) what dialogue control facilities are needed for a p-connection and whether it is appropriate to relay in a transparent way the session service facilities of this type.

2.1.7 Summary Table of Service Primitives for GPS

This table lists the service elements which are included in the GPS Service description. A specific presentation service will use the items in the list relevant to the Facilities in its chosen subset of GPS and can add further service specific service elements; both the choice made and the description of new services must be given in the individual standard.

FACILITY & SERVICE ELEMENTS	TYPE	PRIMITIVES
Presentation-connection Establishment facility		
P-CONNECT	2 RC	request indication response confirmation
Presentation-connection Termination Facility		
P-RELEASE	2 RC	request indication response confirmation
P-DISCONNECT	1 RI	request indication
P-ABORT	3 II	indication
Enclosure Control Facility		
P-CHANGE-ENCLOSURE	2 RC	request indication response confirmation
P-CLOSE/ABORT-ENCLOSURE	2 RC	request indication response confirmation
Negotiation Facilities (see additional information)		
P-PERFORM-NEGOTIATION	2 RC	request indication response confirmation
P-NEG-INVITE	1 RI	request indication
P-NEG-OFFER	1 RI	request indication
P-NEG-ACCEPT	1 RI	request indication
P-NEG-REJECT	1 RI	request indication
Transfer Enclosure Control Facilities		
No specific services are included in this Standard.		

Additional information

Single-interaction negotiation provides only the P-PERFORM-NEGOTIATION service primitives. The other primitives listed under Negotiation Facilities are used only within a negotiation-enclosure.

Sub-clause 2.1.8 defines the GPS subsets and the availability of the services in them.

2.1.8 Subsets

The following subsets of the total service elements included in this Standard are defined for purposes of conformance statements.

GP-A:

No Negotiation: a single Presentation Environment is possible and must be selected (i.e. fully defined) at establishment time. Multiple Presentation Environment option is not available.

Available service elements:

P-CONNECT
P-RELEASE
P-DISCONNECT
P-ABORT

GP-B:

Single-interaction Negotiation without Multiple Presentation Environment option. Only one presentation environment is possible but is selectable, negotiable and re-negotiable.

Available service elements:

as GP-A plus:
P-PERFORM-NEGOTIATION

GP-C:

Single- and Multiple- interaction Negotiation without Multiple Presentation Environment option. As subset GP-B but multiple-interaction negotiation allows more complex presentation environments to be more readily defined.

Available service elements:

as GP-B plus:
P-CHANGE-ENCLOSURE (restricted parameters)
P-CLOSE/ABORT-ENCLOSURE (optional service element)
P-NEG-INVITE
P-NEG-OFFER
P-NEG-ACCEPT
P-NEG-REJECT

GP-D:

Single- and Multiple- interaction Negotiation with Multiple Presentation Environment option. All service facilities are available.

Available service elements:

as GP-C plus:
P-CHANGE-ENCLOSURE (full capabilities)

The combination given by subset GP-B plus Multiple PE option is excluded.

Note 1

Apart from the negotiate-token, which is defined in subset GP-D, GPP does not define the use of arbitration mechanisms, but permits these to be defined as required in specific service standards.

2.2 Service Description

2.2.1 Functional Phases of the Generic Presentation Service

This section gives an account of the principal functional states of the Generic Presentation Service, i.e. of the total Presentation Layer functionality concerned with supporting one pair of presentation users during their establishment, use and termination of the Service. (The states of the Presentation Layer due to the operations of other pairs of users of presentation services are, of course, totally independent).

It identifies only the major phases of the Presentation Layer activity and indicates the dialogues which enable the changes between these phases, but does not show the separate states of the two presentation entities or the detailed states during the operation of the dialogues.

It does not cover the sub-phases which can exist within an information handling activity. Collision prevention or effects are not controlled in GPP, but dealt with as specified in specific service standards.

The concept of presentation environment (PE) is needed in the description of the functional phases. A presentation environment (PE) is a self-consistent usable set of parameters of the presentation service, including those parameters specific to the particular presentation service operating within the framework of GPS. This does not imply that all possible parameters of a service need to be in use in a particular PE. No useful information handling activity is possible except in the context of a Defined PE. In the general case a number of PEs can be defined for one p-connection. The PEs of a p-connection cease to exist (within the scope of the Presentation-Layer) when the p-connection is terminated. In general, Defined PEs are set up by negotiation.

2.2.1.1 Phases of Presentation Service Operation

These are as follows, with a brief note on the function:

1. Unconnected: In this there exists only the potential for establishing a useful presentation-connection (p-connection).

2. Establishment Phase: This transient phase establishes a p-connection between two p-users. The definition of an "initial" PE is optional to this phase. Some specific services may have a "default" PE which is implicitly defined by this phase.

3. Null-enclosure: This phase is entered after completion of establishment phase unless there is then a Defined PE, see 4 below. Controlled or default entry can occur at other times.

4. Transfer Enclosure: This will be implicitly entered after completion of establishment phase when a Defined PE exists at this time according to the parameters of the P-CONNECT service element. It is entered at other times in a controlled and agreed manner. If the multiple-pe option has been agreed at establishment time selection of the PE to be used, from any set of defined PEs which exist, is made at the time of such controlled entry to transfer-enclosure. The PE in use in transfer-enclosure is known as the Current PE; there is no Current PE when transfer-enclosure is not active.

5. Single Interaction Negotiation: This is a transient phase (consisting of a single confirmed service element) which can be entered from either null-enclosure or transfer-enclosure; it always exits to the enclosure from which it was entered. Its purpose is to agree a Defined PE; in the case of entry from transfer-enclosure this will usually imply some re-negotiation of the Current PE, but, where multiple-pe option is in use, additions to the set of Defined PEs can be made in this way.

6. Negotiation Enclosure: This is entered only in a controlled and agreed manner. The purpose of a negotiation enclosure is to agree a Defined PE; a PE so defined may be added to the set of PEs only if multiple-pe option has been agreed, otherwise only negotiation or re-negotiation of the single PE then available is possible.

7. Disestablishment Phase: This phase disestablishes the p-connection, thus losing from the scope of the

p-service all the parameters (PEs) and other information used during the life of the p-connection (any retention of any part of such information by either or both of the p-users is outside the scope of this Presentation-Layer standard).

A diagram illustrating the above is given below, together with some explanatory notes.

Note 2

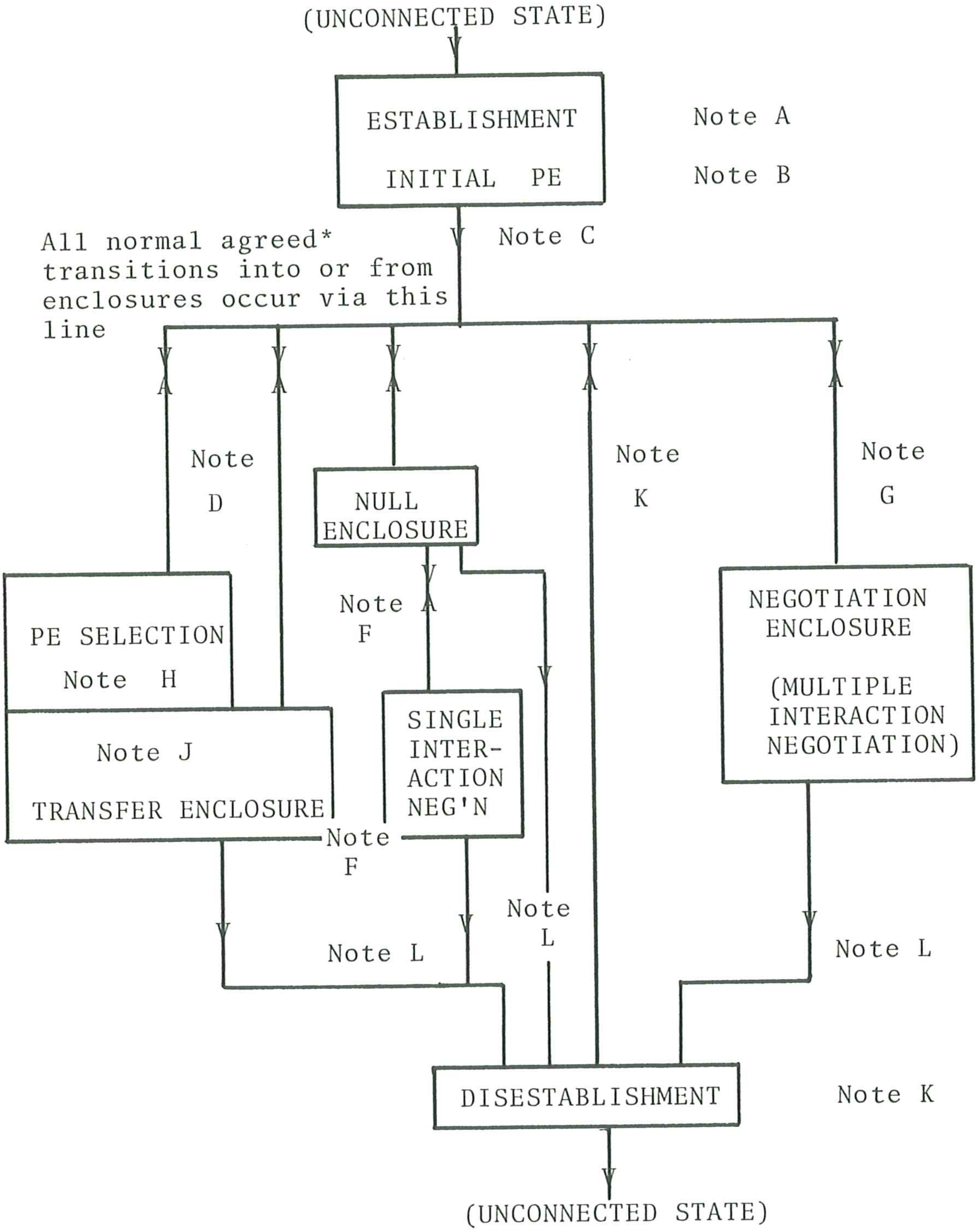
The definition of presentation environments can be based on the concept of profiles which are "well-known" pre-determined sets of parameters. Such profiles will generally be defined either in, or supplementary to, standards for specific presentation services. The maintenance of (and access to) "libraries" of profiles is an OSI management aspect outside the scope of this Standard.

2.2.1.2 Phase Diagram of Presentation Service

The diagram in Fig. 2 illustrates the Phases as identified above. It does not show explicitly the events causing transitions; in general a phase change takes place due to the initiation by one p-user of a valid and successful service element, but in some cases, as explained in the notes following the diagram and in later more detailed descriptions, can occur due to an only partially successful service element. The p-service may force transitions under exceptional circumstances.

This phase diagram is given as an aid to understanding the global phases of activity of a p-service in a pair of interconnected systems. It does not represent the states of any clearly definable component (a more elaborate diagram showing inter-related states of both service-ends would be needed to do this). The "events" causing the "phase-transitions" at this global level are generally complex sequences of activities. The "actions" taken during the transitions may also be complex sequences of activities. If a phase-transition in this diagram is in process and anything in the complex sequence goes wrong a "reset" to the original global phase must occur or some recover/abort to some other phase (transitions the diagram does not attempt to show) must occur.

Further notes on the phase diagram follow the diagram. In this diagram flow is into the top of a phase box and from the bottom, except where indicated by arrows.



* i.e. excluding DISCONNECT

Figure 2. Phase Diagram of VTS

Explanation of GPS Phase Diagram

The letters below refer to the notes in Figure 2.

A. Actions in Establishment Phase cover any necessary activity at the receiving end of the "connect request" to determine whether or not the request can be accepted as well as the action necessary to establish the session-connection with the required characteristics. This may include agreement on the specific presentation protocol required for the p-connection, and the choice of options (subset). (If the establishment attempt does not succeed its effects are lost and the unconnected state is resumed; this transition is uninteresting and is not shown).

B. Definition of all or part of an initial PE is optional in the GPP but may be implicit in some specific p-services. A full PE should be defined for a p-connection for which the negotiation facility option is not used (otherwise no useful work can (ever) be performed). Some (or all) of the initially declared parameters can be "fixed" for the duration of the p-connection.

C. After this transition connection (both session-connection and presentation-connection) is established; all necessary housekeeping associated with this must have been done and the associated information recorded; there may be but will not necessarily be a complete, usable, presentation environment (PE).

D. When establishment has been successful the TRANSFER-ENCLOSURE phase is entered immediately if there is a fully defined PE.

E. NULL-ENCLOSURE is entered implicitly from Establishment phase if there is no fully defined PE. Subsequently it will be entered if there is an explicit request to do so, if an attempt to change from one enclosure to another enclosure is only partly successful (in that only the exit is successful), or in the event of an enclosure being aborted.

F. SINGLE-INTERACTION NEGOTIATION is a phase consisting only of a single confirmed service element entered from either Null-enclosure or Transfer-enclosure (not valid in negotiation-enclosure) and always exiting to the enclosure from which it was entered. Its use is subject to the current GPS subset (which may be changed by negotiation, if available, during the p-connection). Although this phase may be able to change all the characteristics of the PE in use (i.e. the Current PE) the PE ident cannot be changed by the interaction.

G. The use of NEGOTIATION-ENCLOSURE (for multiple-interaction negotiation) is subject to the current GPS subset (which may be changed by negotiation, if available, during the p-connection). Entry to and exit from negotiation enclosure is always by a confirmed service element and there is no "penetration" of previous activity into or after the enclosure. A particular instance of negotiation-enclosure defines (if successful) one PE; this may replace an existing PE or be added as a new PE if multiple-pe option is in use, see note H. Exit may be to null-enclosure, to transfer-enclosure (see notes H,J) or re-entry to negotiation-enclosure.

H. The use of multiple-pe option which enables PE selection is subject to the current GPS subset (which may be changed by negotiation, if available, during the p-connection). This option is needed if more than one separately identified PE is to be used during the life of the p-connection. The maximum number of PEs can be explicitly agreed at establishment time or can be left indefinate/implementation dependent. Selection of a Defined PE to be the Current PE is made at entry to transfer-enclosure and cannot be changed during the transfer-enclosure (the parameters of the PE can be changed by negotiation subject to certain constraints which may have been applied at establishment time).

J. Entry to transfer enclosure is always by a confirmed service element (including the case of implicit entry as a result of establishment, see note D) and there is no "penetration" of activity outside an enclosure. Entry to Transfer-enclosure is possible only if the selected (or only) presentation environment is complete. Facilities available in a presentation environment are class dependent. There may be a hierarchy of activities. A general description is given elsewhere in this document. Termination of transfer enclosure is also normally by a confirmed service element (including the case of implicit termination by Disestablishment of the p-connection) but there is provision for a unilateral forced Close/Abort (and unilateral Disconnect or Abort of the p-connection will also close the transfer-enclosure). Termination of a Transfer Enclosure removes from the current scope of the p-service all the variable information generated during the enclosure, but this information may be held in a way in which it can be restored for further use with the same PE (not in the case of forced termination). The PE itself is not lost and can be re-used for a new presentation environment (not if the p-connection has been broken). Resumption of operation as above is not available in all presentation services or all classes of service.

K. RELEASE of the p-connection is normally initiated from the NULL-ENCLOSURE phase, but can be initiated from within transfer-enclosure. The current set of PEs are lost if RELEASE is agreed but are retained if it is not agreed. All clean-up, freeing of resources, normal forcing of outstanding actions and termination operations must be effected.

L. A request from either p-user to DISCONNECT the p-connection, or a condition causing the p-service to ABORT it, will also force exit from any current enclosure(s) without, in general, the normal completion of current activities.

2.2.2 Establishment of Presentation-Connection

Establishment is concerned with setting up a p-connection between the initiating p-user and a designated peer p-user (with that p-users agreement) and, where desired, agreement of other parameters of presentation service; these may be all or only part of the total set of presentation parameters needed for the communication between the p-users and further negotiation may be needed after establishment. Some of the parameters which are selected at establishment time can be "fixed" for the duration of the p-connection (for all PEs used on the p-connection).

Establishment is also concerned with establishing a session-connection with the appropriate characteristics to support this Presentation Layer activity; some aspects of the session service may be of direct interest to the p-user rather than to the Presentation Layer functions as such. This is subject to definition in specific service standards.

With this version of GPS/P, due to current session service constraints, no change of session connection characteristics can be made subsequent to establishment, and must therefore be "fixed" as above. A later version, subject to enhancement of the session service, may allow for variability in the session characteristics during the lifetime of a particular presentation-connection; for example, it may not be possible always to determine the full session characteristics required for operation of a presentation environment until the negotiation of it is complete. Changes in session characteristics will not be permitted while transfer-enclosure is in progress.

A single service element is provided for establishment with provision for specific service dependent parameters.

The GPS/GPP subset is not inherently fixed following establishment, but can be by agreement; see Additional information to P-CONNECT.

P-CONNECT

Purpose:

To request establishment of a p-connection between the initiating p-user and a peer p-user.

Structures:

Confirmed, type 2, RC.

Parameters:

Parameter Name	req	ind	resp	conf
p-user-caller	D	U	x	x
p-user-called	D	U	D	U
p-fixed-pe-params	D *	U *	D *	U *
p-initial-pe-params	D *	U *	D *	U *
p-max-pes	D	U	D	U
p-save-pe-capab	D	U	D	U
p-transp-data	D	U	D	U
p-diagnostic	x	x	D,U (n)	U

* see notes on these parameters below.

Parameter descriptions:

p-user-caller: Optional: if present, gives the name of the caller and is passed in the indication (some p-users may refuse to accept "anonymous" callers).

p-user-called: The name of the called p-user. Optional in response and confirmation primitives.

p-fixed-pe-params: Optional: enables p-service and class, p-service/class-versions and further specific service dependent parameters to be agreed and fixed at establishment time; any parameters agreed by means of this parameter cannot be changed during the subsequent life of the p-connection. The sub-parameters may, but will not necessarily, include a "pe-profile" (which is specific service dependent in nature). The PE defined, wholly or in part, at establishment time is known as the "initial PE" (it has pe-ident value "0"). Further information on component sub-parameters is given below.

p-initial-pe-params: Optional: enables p-service and class, p-service/class-versions and further specific service dependent parameters to be agreed at establishment time for initial use; any parameters initially agreed by means of this parameter are subject to agreed variation by re-negotiation during the subsequent life of the p-connection. The sub-parameters may, but will not necessarily, include a "pe-profile" (which is specific service dependent in nature). The PE defined, wholly or in part, at establishment time is known as the "initial PE". Further information on component sub-parameters is given below.

p-save-pe-capab: Optional: determines whether the service provides a PE saving capability or not. Symbolic values are "yes", "no".

p-max-pes: Optional: enables the maximum number of Defined PEs to be agreed. Value is numeric. Response may not contain a value greater than in the indication. If not used (absent from the response/confirmation primitives) the number of PEs is assumed effectively unlimited, unless defined by the choice of a specific service/class standard.

p-transp-data: Optional: information in this parameter supplied by p-user is conveyed as a transparent binary octet string by the p-service. See Additional information.

p-diagnostic: Advises the result of the service element, see 2.2.10 for further information. Minimum size is 4 octets, see Additional information.

Component parameters of p-fixed-pe-params and p-initial-pe-params:

The following parameters can be used in these compound parameters and are then used as shown in the following table and accompanying notes:

Parameter Name	req	ind	resp	conf
p-service-ident	D	U	x	x
p-service-class-ident	D	U	x	x
p-service/class-version-idents	D (n)	U	D (1)	U
p-service/class-subset-ident	D (n)	U	D (1)	U
p-pe-profile	D	U	D	U
p-special-conventions	D (n)	U	D (n)	U

Parameter descriptions:

p-service-ident: Applicable symbolic values are "DPS", "VTS"; other values are reserved. If the "offer" in the request/indication is not acceptable to the p-user (or the p-entity) the connection request will fail.

p-service-class-ident: This is applicable, but optional, if a specific service is selected and enables the class of this service to be agreed if there are choices of class. Applicable values are specific service dependent. If the "offer" in the request/indication is not acceptable to the p-user (or the p-entity) the connection request will fail.

p-service/class-version-idents: This is applicable if p-service or, where relevant, p-service-class is selected and enables the service or class version (and hence protocol version) to be agreed if there are choices of version. Applicable values are numeric as given in specific service standards. The response primitive may contain only one of the values given in the indication primitive. The p-entity itself may discard some values given in the request primitive before giving the indication primitive. If there is no acceptable value the connection attempt will fail.

p-service/class-subset-ident: This is applicable, but optional, if a p-service or p-service-class is selected and enables the subset to be agreed if there are choices of subset. Applicable values and rules for values in request and response are as given in specific service standards; it is expected that the subsets will usually form a hierarchical ordered set and that the response will be allowed to contain a value equal to or lower in this order than in the request, otherwise the connection attempt will fail.

p-pe-profile: This is applicable, but optional, if p-service, p-service-class as appropriate, and p-service/class-subset as appropriate, are selected, and enables a full p-service profile to be agreed at establishment time; the nature is service/class/subset dependent. The value consists, in general, of a profile-ident plus, if relevant, profile-parameters, as defined in specific service standards. If the profile offered is not acceptable to the p-user (or the p-entity) the connection attempt will fail. If the profile is accepted the transfer-enclosure will be entered when establishment is complete. Applicability of return values is service and class dependent.

p-special-conventions: Optional: specifies which special presentation conventions may be used on this p-connection. Agreement allows use of p-special-information parameter in certain other service elements. The meaning of these parameters is user defined. It may be appropriate for the response to contain more than one return value but these must all have been included in the indication.

A number of the parameters are of interest to the p-service; if a value in request or response is not acceptable to the p-service an appropriate diagnostic will be generated; in the case of rejection by the p-entity there may be no indication primitive (and hence no response primitive); p-transp-data in the request primitive is discarded in this case.

Usage and Effects

Request primitive can be issued by a potential p-user at any time. Effects are sequenced. No further request primitive will be accepted from the initiating p-user until the confirmation primitive has been issued, except P-DISCONNECT request primitive. No primitive from the response issuer will be allowed to overtake the confirmation except P-DISCONNECT which will destroy the effects of P-CONNECT and of any other issued request primitives. A P-ABORT will disrupt a P-CONNECT action and any other service element in progress.

If all parameters present are acceptable to the p-service and the responding p-user the p-connection is established. If a full profile (PE) has been selected, transfer enclosure will be entered, otherwise null-enclosure is entered.

Additional information

p-user-caller and p-user-called will not necessarily retain their values unchanged between the two p-users (session service dependent, see ECMA-75). No explicit parameter is included for the GPS/GPP subsets as defined in clause 2.1.8 of this Standard since the GPS subset is selected implicitly by the specific service/class/operating-subset parameters as defined in specific service standards. There is no inherent restriction in this Standard on a change to this GPS subset during the life of a p-connection if a change of the operating-subset, class or service so demands.

With this version of GPS/GPP, parameters as above which affect the selection of session service facilities must be given in p-fixed-pe-params, and are selectable between request and response only to the extent available in session connection establishment, see ECMA-75.

If parameters included in p-initial-pe-params are to be changed during the life of the p-connection it must be ensured that a negotiation facility is available in the choice of service/class/subset.

If a complete p-service profile (PE) is not selected at establishment time (i.e. the parameters in the above table are not all present in p-fixed-pe-params or p-initial-pe-params taken together) and negotiation facility is not available no further useful operation of the p-connection is possible; P-DISCONNECT or P-RELEASE should be performed. A specific service standard may "trade off" octets between p-transp-data and p-fixed-pe-params and/ or p-initial-pe-params; the total sum length of these parameters is restricted to 50 octets; this figure is subject to review. The use of p-diagnostic longer than the minimum issued in response primitive (D case) is permitted but must be covered by the same trade-off.

2.2.3 Termination of Presentation Connection

Termination is concerned with closing down the presentation service as far as the two connected p-users are concerned, either with mutual agreement or unilaterally. In some specific services a token may be defined for mediating use of agreed termination.

The following service elements are available, supporting the three forms of this facility.

P-RELEASE

Purpose:

To request orderly termination of the p-connection between the initiating p-user and a peer p-user.

Structures:

Confirmed, type 2, RC.

Parameters:

Parameter Name	req	ind	resp	conf
p-transp-data	D	U	D	U
p-diagnostic	x	x	D(n)	U

Parameter descriptions

p-transp-data: Optional: information in this parameter supplied by p-user is conveyed as a transparent binary octet string by the p-service. Maximum length is 24 octets.

p-diagnostic: advises the result of the service element, see 2.2.10 for further information. Minimum size is 4 octets, see Additional information.

Usage and Effects

May be issued by either p-user subject to any rules in the p-service and/or service-class being used for the p-connection; this may include a token for mediating issue of request primitive. Effects are sequenced and non-destructive. No other request primitive will be accepted from the request issuer until the response/confirmation has been given, but further indication primitives may be given to the request issuer due to service elements initiated by the peer p-user before the P-RELEASE indication primitive was given to the peer p-user.

If the Release is successful, p-diagnostic severity is "success", the p-connection is broken and all Defined PEs are lost. The session-connection is also broken. If the p-diagnostic severity is other than "success" the previous state of the p-connection is restored and requests will again be accepted according to the normal rules of the p-service in operation.

Additional information

The use of p-diagnostic longer than the minimum issued in response primitive is permitted but subject to trade off with p-transp-data.

P-DISCONNECT

Purpose:

To force termination of the p-connection between the initiating p-user and a peer p-user.

Structures:

Non-confirmed, type 1, RI.

Parameters:

Parameter Name	req	ind
p-transp-data	D	U

Parameter descriptions

p-transp-data: Optional: information in this parameter supplied by p-user is conveyed as a transparent binary octet string by the p-service. Maximum length is 31 octets.

Usage and Effects

May be issued by either p-user at any time; no service or class is permitted to place any restriction on the issue of this service element. Effects may be expedited and destructive. The p-connection is broken and the s-connection disconnected also. This implies forced termination of any enclosure in use and discard of its contents. All Defined PEs are lost.

Additional information

In the event of a collision of P-DISCONNECT request primitives either or both indication primitives may be lost but the disconnection as above will always occur.

P-ABORT

Purpose:

Generated by the p-service to indicate that a p-connection has been lost due to p-service or lower layer service exception condition.

Structures:

Indication only, type 3, II.

Parameters:

Parameter Name	ind	ind
p-diagnostic	U	U

Parameter descriptions

p-diagnostic: Indicates the reason for the loss of p-connection. See 2.2.10 for further information. The two presentation service users may receive different diagnostics.

Usage and effects

This service element may be generated by the p-service at any time. Effects are destructive. The p-connection has been broken and the s-connection is disconnected if this is not already lost (by S-ABORT). All parameters of the p-connection are assumed lost, and no further indications or confirmations will be received for this p-connection. This implies forced termination of any enclosure in use and discard of its contents. All defined PEs are lost.

2.2.4 Enclosure Control Facility

A single service element as below is provided to effect the controlled change of context from one form of enclosure to another or re-entry to the same form of enclosure with some different (or the same) entry parameters. Thus this service element can make the transition between any pair of transfer-enclosure, negotiation-enclosure and null-enclosure including pairs of transfer-enclosures and of negotiation-enclosures (pairs of null-enclosures are not excluded but are not useful); negotiation-enclosure is not always available, see clause 2.1.8.

P-CHANGE-ENCLOSURE

Purpose:

To request orderly exit from the current enclosure of whatever type and entry to a new designated enclosure of the same or other type.

Structures:

Confirmed, type 2, RC.

Parameters:

Parameter Name	req	ind	resp	conf
p-dest-encl-type	D	U	D	U
p-exit-params	D	U	x	x
p-entry-params	D	U	x	x
p-transp-data	D	U	D	U
p-diagnostic	x	x	D,U (n)	U

Parameter descriptions

p-dest-encl-type: Designates the type of enclosure which is to be entered if all the necessary conditions are satisfied, see parameters below. Applicable values are "null", "transfer", "negotiation". The value in the confirmation primitive gives the type of enclosure in operation following the service element (which may be unchanged, see Effects and additional information). "Negotiation" is valid only if use of negotiation-enclosure has been agreed (GP-C or GP-D subsets, see 2.1.8).

p-exit-params: This contains parameters relating to the exit conditions of the current enclosure, i.e. that in which the peer p-users are operating at the time when the service element is initiated. If the exit conditions are not acceptable to the receiving p-entity or the associated p-user the service element will completely fail with p-diagnostic giving the reason; see also Effects below. The nature of p-exit-params is dependent on the type of the current enclosure as given below and this parameter is not relevant if the current enclosure is the null-enclosure.

p-entry-params: This contains parameters relating to the entry conditions to the requested destination enclosure. If the entry conditions are not acceptable to the receiving p-entity or the associated p-user the current enclosure will be closed in good order (assuming p-exit-params were acceptable as above) but the null-enclosure will be entered instead of the designated destination enclosure; see also Effects below. The nature of p-entry-params is dependent on the type of the destination enclosure designated by p-dest-encl-type as given below and this parameter is not relevant if the intended destination is the null-enclosure.

p-transp-data: Optional: information in this parameter supplied by p-user is conveyed as a transparent binary octet string by the p-service. Maximum length is 48 octets.

p-diagnostic: Advises the result of the service element, see 2.2.10 for further information, also Effects below.

p-exit-params

Exit from Negotiation-enclosure:

Relevant exit parameters, applicable values and consequent actions are described in 2.2.7.2.

Exit from Transfer-enclosure:

Relevant exit parameters, applicable values and consequent actions are described in 2.2.8.4.

p-entry-params

Entry to Negotiation-enclosure:

Relevant entry parameters, applicable values and consequent actions are described in 2.2.7.1.

Entry to Transfer-enclosure:

Relevant entry parameters, applicable values and consequent actions are described in 2.2.8.3.

Usage and Effects

The request primitive may only be issued by the current holder of the enclosure-token or, if this token is not defined, the initiator of the p-connection. The effects are sequenced and non-destructive. No other request primitive except P-DISCONNECT request will be accepted from the request issuer until the response/confirmation has been given but further indication primitives may be given to the request issuer due to service elements initiated by the peer p-user before the P-CHANGE-ENCLOSURE indication was given to the peer p-user; this p-user will not be permitted to issue further requests following this indication until after the response primitive; normal rules for the service condition then assumed to exist then apply. The possibility and effects of collision with other service elements, in cases with multiple tokens, will be defined, when relevant, in specific service standards. A later version of this Standard may give general rules.

Unless the exit conditions are unsatisfactory (not applicable if the current enclosure is the null-enclosure) and if the peer p-user agrees, the current enclosure is closed in an orderly manner. Otherwise the current enclosure is retained and the p-entities are restored to the states before the request was issued. If exit does occur, then, unless the entry conditions are unsatisfactory (not applicable if the designated destination is the null-enclosure) and if the peer p-user agrees the new enclosure is entered with the designated entry conditions. The p-dest-encl-type parameter in confirmation primitive always shows the type of enclosure in operation after the service element whether this has been wholly unseccessful or partially or wholly successful; p-diagnostic gives further information on occurrence of and reason for partial or complete failure. Severity value will be "success with warning" if exit occurs but entry is not possible.

Additional information

Ensuring that the responding p-user is aware of the final state if this is different to that in the response (due to an error in the response) is a local implementation matter outside the scope of this Standard. The enclosure-token, if defined, always remains with the issuing p-user at the conclusion of the service element.

P-CLOSE/ABORT-ENCLOSURE

Purpose:

To force the termination of the current enclosure and entry to the null-enclosure. The effect on actions in progress in the closed enclosure is dependent on its type.

Structure:

Confirmed, type 2, RC.

Parameters:

Parameter Name	req	ind	resp	conf
p-transp-data	D	U	D	U

Parameter descriptions

p-transp-data: Optional: information in this parameter supplied by p-user is conveyed as a transparent binary octet string by the p-service. Maximum length is 48 octets.

Usage and Effects

The request primitive may be issued by either p-user independently of the possession of a token. This service element may not be available in all services and classes. Effects are sequential and potentially destructive, see Additional information. For the effect on particular types of enclosure (abort of null-enclosure is not interesting) refer to description of these enclosures in later subclauses.

Additional information

There can be collision between this service element and other service elements as below; the effects are as noted:

1) service elements used in the enclosure being closed; in this case this CLOSE/ABORT-ENCLOSURE service element is the contention winner and parts of the colliding service elements may be lost.

2) P-DISCONNECT service element; in this case the P-DISCONNECT is the contention winner and parts of the P-CLOSE/ABORT-ENCLOSURE may be lost.

3) P-CLOSE/ABORT-ENCLOSURE service element; in this case the CLOSE/ABORT-ENCLOSURE issued by the connection initiator is the contention winner and parts of the service element of the contention loser may be lost.

2.2.5 Negotiation Facility

2.2.5.1 Introduction

A service of the Presentation Layer cannot perform a useful function for the p-users until a complete and self-consistent set of values of presentation parameters has been established; such a set is known as a Presentation Environment (PE) and no information (other than, possibly, "transparent data") can be handled until at least one PE exists in Defined state. Depending on the GPP Service-subset it may be possible to have a number of PEs defined at any one time. An initial PE can be Defined during establishment of the p-service. If this "initial" PE is only partially declared at establishment time it can be completed later by negotiation, and subsequently changed if necessary.

The General Presentation Service provides, as an optional facility negotiated at establishment of the p-service (by means of p-service-subset), Negotiation Facilities which enable one or more presentation environments to be declared, negotiated or re-negotiated by the two p-users. The Negotiation Facilities in the service are concerned with any parameter in the presentation layer which affects the p-users usage of the Service.

Multiple PEs are relevant only if use of the Multiple-PE option is agreed. The maximum number of PEs can be agreed; otherwise it is assumed to be effectively unlimited. If Multiple-PE option is not in use only the "initial" PE can exist but its parameters can be changed from time to time by re-negotiation (subject to constraints placed on such changes at establishment time - and to use of negotiation being itself agreed).

2.2.5.2 The Negotiation Facilities

Two forms of negotiation are available. The first is a Single-interaction Negotiation in which a negotiation consisting only of a single exchange is performed by a specific Confirmed service element.

The second, more complex form, is the Multiple-interaction Negotiation which is always performed within a

Negotiation-enclosure, which is entered and terminated by confirmed service elements. Any one negotiation-enclosure is concerned with one specific presentation environment (PE). While a negotiation-enclosure is in progress the PE being developed is known as a Draft Presentation Environment. A Draft PE will become a Defined PE if the negotiation-enclosure is successfully terminated and is descarded otherwise. Some Services and Classes may allow for only one PE and in such a case only negotiation or re-negotiation of this single PE (the "initial" PE) is possible.

A Negotiation-enclosure can be used to re-negotiate an existing PE; this PE is retained but a copy is also taken into the Negotiation-enclosure to form the Draft PE for the enclosure. Depending on the outcome of the enclosure the old PE may be replaced by the new one, may remain in existence, or may be deleted.

GPP provides two levels of Negotiation Facility; the Single-interaction form only, as available in subset "GP-B", and the full facility, available in subsets "GP-C" and "GP-D", giving also the Multiple-interaction form.

2.2.5.3 Concepts of Negotiation Process

Each Presentation Environment (PE) is defined by a complete and self-consistent set of presentation parameters. The Negotiation Process allows the service users to modify an existing PE, to create a new PE or to delete an existing PE. Within a set of parameters certain relationships exist which may be expressed as a Directed Graph. These relationships determine the order in which parameters may be negotiated. Each version of each class of each presentation service has a directed graph of parameters. Certain higher order parameters may have values which are fixed at presentation connection time and this determines the valid entry points into the graph at which negotiation may begin. Lower order parameters may not be applicable by reason of higher order parameters.

The multiple-interaction negotiation process is controlled by the negotiate-token. The holder of the token may start negotiation for any of the set of parameters, subject to the constraints of the parameter relationships and any parameters which have fixed values. When the negotiate-token is passed, directly related parameters for which negotiation has been started since the preceding token passing are linked into a negotiation group, all parameters in each group being in the same state. The highest order parameter in a group is called the root parameter and any parameter may be a member of one and only one negotiation group at any one time. No parameter may be subsequently added to the group. Other directly related parameters, parents or children, become "frozen" and remain in that state until the negotiation cycle for the group completes (when the group ceases to exist).

At each exchange of the token the states of the parameters in each negotiation group are examined and the following rules applied. Individual parameter states may have changed between token exchanges, (see parameter state table):

i) Parameters which have been accepted remain in that state. The children of parameters, which have been rejected and are now "frozen", become "frozen".

ii) All other parameters and their children within the negotiation group which do not have the same state as the root parameter become "frozen" and are no longer regarded as part of the negotiation group.

iii) When the root parameter is agreed or rejected the negotiation group ceases to exist on the first subsequent passing of the negotiate-token.

iv) When the negotiation group ceases to exist, parents return to their original state and value unless any other children are involved in another still existing negotiation group.

v) If the root parameter was rejected, all children of the group return to their original states and values. If a new value for the root parameter has been agreed, all children are set to either a "not applicable" state or an "agreed/negotiable" state with an appropriate default value.

		OUTSIDE NEGOTIATION ENCLOSURE	EXTERNAL	NOT APPLICABLE	AGREED/ NEGOTIABLE	INVITED	OFFERED	COUNTER-OFFERED	FROZEN
STATE		0	1	2	3	4	5	6	7
EVENT									
0 CHANGE ENCL to NEG'N/ PERFORM NEG		D 1 2 3							
1 INVITE					4				
2 OFFER					5	5	6		
3 ACCEPT							3	3	
4 REJECT							7	7	
5 TOKEN PASSED	PofG		1		7				A 3 | 7
	MofG				3	B 4 | 7	B 5 | 7	B 6 | 7	
	CofG			7	7				C 2 | 3
	ind		1	2	3	4	5	6	7
6 CHANGE ENCL from NEG'N "SWITCH"			0	0	0	4 change fails	5 change fails	6 change fails	7 change fails
7 CHANGE ENCL from NEG'N "DELETE" or "RESTORE" or CLOSE/ ABORT			0	0	0	0	0	0	0

For Key and Notes (letters in intersections) see next page.

Key:		
	P of G	- Parent of Group
	C of G	- Child of Group
	M of G	- Member of Group
	ind	- independent parameter

Notes on Parameter State Table

A. Parent state changes to "agreed/negotiable" only when negotiation group ceases to exist and no other children are involved in other nagotiation groups.

B. When a member of a group has a state which is different to the root member's state, it and it's children's states change to "frozen" and drop out of the negotiation group.

C. Depending on the value of higher order parameters the frozen state will be either "not applicable" or "agreed/negotiable" with an appropriate default value.

D. A parameter will, if fixed at connection time, have an "external" state. A parameter will have a "not applicable" state if not applicable by reason of higher order parameter values. Otherwise a parameter will have an "agreed/negotiable" state.

2.2.6 Single-interaction Negotiation

The following service element is available when use of Negotiation Facility has been agreed, i.e. the current subset is other than "GP-A". It can be used in null-enclosure or in transfer-enclosure. No change of enclosure is ever made by this service element and it cannot change the PE-ident in use in transfer-enclosure (although it may be able to change all other parameters of this PE and may establish a new but inactive PE).

P-PERFORM-NEGOTIATION

Purpose:

To request the performance of a single-interaction negotiation.

Structure:

Confirmed, type 2, RC.

Parameters:

Parameter Name	req	ind	resp	conf
p-pe-ident	D	U	x	x
p-draft-pe	D	U	x	x
p-pe-specific-params	D (n)	U	D (1)	U

Parameter Name	req	ind	resp	conf
p-special-info	D (n)	U	D (n)	U
p-transp-data	D	U	D	U
p-diagnostic	x	x	D,U (n)	U

Parameter descriptions

p-pe-ident: Determines the intended action of the negotiation. Optional, and Invalid if multiple PE option is not in use. Symbolic values are "initial", or "pe-ident/n" the designated PE may be the Current PE. If the designated PE exists the intended action is to replace its definition with the result of the negotiation; if it does not exist the intended action is to generate a new PE defined according to the result of the negotiation. See 2.2.8.2 for further information on values of pe-ident/n. The default if the parameter is not present is "initial".

p-draft-pe: Determines the initial values of the parameters for negotiation. Optional, and Invalid if multiple PE option is not in use. Symbolic values are "initial", which commences the negotiation with all parameters set as currently defined in the "initial" PE (which may be "non-existing"), or "pe-ident/n", which causes this designated (already existing) PE to be copied as the draft PE on which negotiation is commenced; the designated PE may be the Current PE. See 2.2.8.2 for further information on values. The default if the parameter is not present is "initial".

The above parameters p-pe-ident and p-draft-pe are not negotiable and if the values in the request/indication are not acceptable to the p-user or the p-entity the service element will fail.

p-pe-specific-params: To complete, as necessary, the set of parameters used in the negotiation. Can contain subparameters as given for p-initial-pe-params under P-CONNECT; any not included will retain values as implied by p-draft-pe (which will include implicitly any values fixed at establishment time). Can also contain further specific PE parameters. In the logical order (p-draft-pe), (p-pe-specific-params), logically later values will override logically earlier values according to the (service/class specific) rules for PE parameter negotiation. The set of parameters supplied by the request issuer need not make up a fully defined PE and can contain alternatives but the set returned by the response issuer must make up a fully defined (usable) PE with all choices made, if the negotiation is to succeed. The rule about "fixed" parameters as given above is applicable. There is no means, in the single-interaction negotiation service element, for the request issuer to reject the result of such choices left open to the response issuer. See Additional information.

p-special-info: User-defined information conforming to the special conventions agreed at establishment time.

p-transp-data: Optional: information in this parameter supplied by p-user is conveyed as a transparent binary string by the p-service. Maximum length is 48 octets.

p-diagnostic: Advises the result of the service element, see 2.2.10 for further information.

Usage and Effects

Request primitive may be issued by either p-user. If the issue is not mediated by a token (specific-service-dependent) and collision occurs the request of the connection initiator is the contention winner. Effects are sequenced and non-destructive except when collision occurs as above. Reception by an issuer of P-PERFORM-NEGOTIATION request primitive of a P-PERFORM-NEGOTIATION indication primitive implies that contention has been lost and the request primitive has been discarded. Subject to token-control as above and also to service-specific restrictions this service element can be initiated while transfer-enclosure is in progress and will transiently suspend the activity with no penetration across the single-interaction negotiation. The (successful) request issuer is not permitted to issue any further requests (except P-DISCONNECT) until the confirmation is received but must be prepared to receive further indications. The peer p-user is not permitted to issue further requests (except P-DISCONNECT) after receipt of the indication until the response has been sent.

If negotiation is successful the intended action implied by the value of p-pe-ident is obeyed. If the Current PE has been altered by the negotiation when entered directly from transfer-enclosure as above, the new form will be used when transfer activity is resumed after the response/ confirmation primitives. If a new PE is generated or an inactive PE is amended, this PE is stored but is not activated; the Current PE is resumed as though the service element had not been performed. If negotiation has been unsuccessful all effects of the service element are annulled and the state restored as though the request had never been issued. This service element never changes the type of enclosure in operation even when it "fails".

Additional information

The condition for success or failure of negotiation and the effects of successful negotiation on any information held with the presentation environment is dependent on the service and class and on the parameters being negotiated and will be given in service specific standards. The p-entity associated with the responding p-user will give a failure diagnostic (U case) to this p-user if this p-entity considers that the PE as set up by the response parameters is not satisfactory. It is not permissible to attempt to change any of the parameters which were "fixed" at establishment time. Inclusion of any such parameter with a different value implicitly or explicitly through p-pe-specific-params, will result in the service element failing with appropriate diagnostic information.

2.2.7 Multiple-interaction Negotiation and Negotiation Enclosure

With this form of negotiation the P-CHANGE-ENCLOSURE service element is used to formally open and terminate the negotiation enclosure with further service elements being available for performing the negotiation within the enclosure.

2.2.7.1 Entry to Negotiation Enclosure

The conditions of entry to the negotiation-enclosure, which will determine the overall success of the P-CHANGE-ENCLOSURE service element as described in 2.2.4 are given by the following enclosure specific information:

p-dest-encl-type: Takes value "negotiation".

p-entry-params: As table and descriptions below.

Entry Parameters for Negotiation-enclosure:

Parameter Name	req	ind	resp	conf
p-pe-ident	D,U	U	x	x
p-draft-pe	D	U	x	x
p-initial-pe-params	D	U	D	U
p-special-info	D (n)	U	D (n)	U

Parameter descriptions

p-pe-ident: Determines the intended action of the negotiation. Optional, and Invalid if multiple PE option is not in use. Symbolic values are "initial", or "pe-ident/n". If the designated PE exists the intended action is to replace its definition with the result of the negotiation; if it does not exist the intended action is to generate a new PE defined according to the result of the negotiation. See 2.2.8.2 for further information on values of pe-ident/n. The default if the parameter is not present is "initial".

p-draft-pe: Determines the initial values of the parameters for negotiation. Optional, and Invalid if multiple

PE option is not in use. Symbolic values are "initial" which selects all parameters as currently defined in the "initial" PE (which may be "non-existing"), or "pe-ident/n" value which causes this designated (already existing) PE to be copied as the draft PE. See 2.2.8.2 for further information on values. The default if the parameter is not present is "initial".

p-initial-pe-params: Applicable when p-draft-pe is "initial"; the component parts of this parameter can be as given under P-CONNECT for p-initial-pe-params: see also Additional information.

p-special-info: User-defined information conforming to the special conventions agreed at establishment time.

Usage and Effects

See description of P-CHANGE-ENCLOSURE for general information on usage and effects.

If the P-CHANGE-ENCLOSURE service element is fully successful as indicated by p-diagnostic, a Negotiation-enclosure is started with the designated parameters. If the receiving p-entity itself rejects the request there may be no indication or response primitive and in this case p-transp-data in the request primitive will be discarded. The responding p-entity and p-user can only accept or reject the proposed entry parameters; in the event of such failure the overall effect of the service element will be entry to the null-enclosure, see description of P-CHANGE-ENCLOSURE.

Additional information

The action within a negotiation-enclosure is mediated by a token, the negotiate-token, and this is initially held by the p-user which issued the (successful) request primitive.

It is not permissible to attempt to change, by means of p-initial-pe-params, any of the parameters which were "fixed" at establishment time. Inclusion of any such parameter with a different value implicitly or explicitly through p-initial-pe-params will result in the entry part of the service element failing as above.

2.2.7.2 Exit From Negotiation Enclosure

A negotiation-enclosure can be terminated with agreement of both p-users by means of P-CHANGE-ENCLOSURE or, subject to service specific restriction, forced unilaterally by either of them by means of P-CLOSE/ABORT-ENCLOSURE.

2.2.7.2.1 Agreed Exit

This uses P-CHANGE-ENCLOSURE with exit parameters, which condition the success of the service element as described in 2.2.4, as given below:

p-exit-params: See table and descriptions below:

Exit Parameters from Negotiation-enclosure:

Parameter Name	req	ind	resp	conf
p-pe-action	D	U	x	x

Parameter descriptions

p-pe-action: takes one of the following symbolic values with intended effects as given under Effects:

- "switch"
- "restore"
- "delete"

Usage and Effects

See description of P-CHANGE-ENCLOSURE for general information. The following is enclosure specific information.

"switch": this action is possible if the Draft PE is in a complete (usable) state; the Draft PE is established as a Defined PE, either a new one or replacing an existing one as implied by the value of p-pe-ident at entry; the negotiation-enclosure is terminated. See Additional information.

"restore": only applicable if p-pe-ident at entry to negotiation-enclosure implied "replace" action; the original PE is restored, the Draft PE is discarded and negotiation-enclosure is terminated.

"delete": the Draft PE is discarded, and, in the case of "replace" action the corresponding existing PE is also deleted, and negotiation-enclosure is terminated.

Additional information

It is necessary for both the p-users and the p-entities to be satisfied with the completeness of a Draft PE before a "switch" can be successful. P-diagnostic is used to convey success or reason for failure. "Restore" or "delete" is always successful. Any information held with a PE due to save-option value "save" in the exit parameters from transfer-enclosure, (2.2.8.4), is lost if the PE is successfully re-negotiated, but is retained if re-negotiation is not successful and the PE is retained unchanged.

2.2.7.2.2 Unilateral Close/Abort

This uses P-CLOSE/ABORT-ENCLOSURE; there are no exit conditions as the service element is always successful, and the null-enclosure is always entered after the completion of the service element.

Enclosure-specific information

When negotiation-enclosure is aborted the PE which was being negotiated adopts the state (which may be non-existing) which it had prior to the entry to the negotiation-enclosure (including any "saved" information). Any service element which had been initiated by either p-user will have been discarded and will result in no primitives subsequent to the confirmation primitive of the CLOSE/ABORT-ENCLOSURE service element. After completion of this service element the enclosure-token, if defined, is owned by the initiator of the aborted negotiation-enclosure.

2.2.7.3 Facilities Within Negotiation Enclosure

The following service elements are available in the general case while a negotiation enclosure is open but some presentation services may not provide all of them due to limited requirements. This will be stated in the relevant standards. No change to any parameters "fixed" at establishment time may be attempted by the parameters of the following service elements.

P-NEG-INVITE

Purpose:

To invite the peer p-user to provide values for a particular parameter or set of parameters and to inform the p-service of this invitation.

Structure:

Non-confirmed, type 1, RI.

Parameters:

Parameter Name	req	ind
p-param-ident-list	D	U
p-transp-data	D	U

Parameter descriptions

p-param-ident-list: Contains one or more presentation parameter identifiers; these must all refer to parameters which are currently in "agreed/negotiable" state. As a default, symbolic value "all" may be used to invite offers for all parameters which are in a negotiable state.

p-transp-data: Optional: information in this parameter supplied by p-user is conveyed as a transparent binary octet string by the p-service. Maximum length is 48 octets.

Usage and Effects

The request primitive may be issued only by the current owner of the negotiate-token. Effects are sequenced and non-destructive. The status of the parameters included in the p-param-ident-list becomes "invited" and the peer p-user should normally issue a P-NEG-OFFER service element in due course (when in possession of the negotiate-token).

When the negotiate-token is next passed, other parameters of the Draft PE may change their status from "agreed/negotiable" to "frozen" in accordance with the Parameter Graph appropriate to the p-service and class of the PE being negotiated.

P-NEG-OFFER

Purpose:

To provide the p-service and the other p-user with a set of values for one or more parameters.

Structure:

Non-confirmed, type 1, RI.

Parameters:

Parameter Name	req	ind
p-param-list	D	U
p-transp-data	D	U

Parameter descriptions

p-param-list: Contains one or more presentation parameter identifiers with proposed values or value ranges; these must all refer to parameters which are currently in one of the following states:

"agreed/negotiable", "invited" or, "offered".

p-transp-data: Optional: information in this parameter supplied by p-user is conveyed as a transparent binary octet string by the p-service. Maximum length is 48 octets.

Usage and Effects

The request primitive may only be issued by the current owner of the negotiate-token. Effects are sequenced and non-destructive. As a consequence of this service element the states of the offered parameters change as follows:

"agreed/negotiable" becomes "offered"
"invited" becomes "offered"
"offered" becomes "counter-offered"

Moreover, if one or more parameters leave the status "agreed/negotiable", some other parameters (not mentioned in p-param-list) may change their status from "agreed/negotiable" to "frozen" in accordance with the Parameter Graph appropriate to the service and class of the PE being negotiated, when the negotiate-token is next passed.

The p-service will check the offer against its own capabilities and modify the offer if necessary.
The receiving p-user should normally either Accept the Offer or Reject the Offer or (if allowed) make a counter-offer.

Additional information

The "counter-offer" facility may not be available in some services and classes as stated in the relevant standards.

P-NEG-ACCEPT

Purpose:

To inform the p-service and the peer p-user of the value (value range) that is accepted for one or more parameters of the Draft PE.

Structure:

Non-confirmed, type 1, RI.

Parameters:

Parameter Name	req	ind
p-param-list	D	U
p-transp-data	D	U

Parameter descriptions

p-param-list: Contains one or more presentation parameter identifiers with one proposed value or value range; these must all refer to parameters which are currently in one of the following states:

"offered" or "counter-offered".

If the state is "offered", the offer must have been made by the other p-user; if the state is "counter-offered", the counter-offer must have been made by the other p-user. The accepted value(s) is (are) specified even if the Offer Indication to which it is a reaction offered only one value for the parameter.

p-transp-data: Optional: information in this parameter supplied by p-user is conveyed as a transparent binary octet string by the p-service. Maximum length is 48 octets.

Usage and Effects

The request primitive may only be issued by the current owner of the negotiate-token. Effects are sequenced and non-destructive. In the Draft PE the parameter(s) adopt the accepted values (value ranges).
If any parameter is negotiated which prohibits this parameter from being negotiated the parameter state becomes "frozen", otherwise "agreed/negotiable".

When the negotiate-token is next passed, or when the negotiation-enclosure is successfully terminated with option "switch", whichever happens first, the applicable Parameter Graph for this service and class indicates actions that may be needed on other parameters.
Examples - parameters that are no longer significant,
         - status of parameter changes from "frozen" to "agreed/negotiable".

P-NEG-REJECT

Purpose:

To inform the p-service and the peer p-user that an offer or counter-offer, made using P-NEG-OFFER, contained one or more unacceptable parameters and/or values.

Structure:

Non-confirmed, type 1, RI.

Parameters:

Parameter Name	req	Ind
p-param-ident-list	D	U
p-transp-data	D	U

Parameter descriptions

p-param-ident-list: Contains one or more presentation parameter identifiers; these must all refer to parameters which are currently in one of the following states:

"offered" or "counter-offered".

If the state is "offered" the offer must have been made by the other p-user; if the state is "counter-offered", the counter-offer must have been made by the other p-user.

p-transp-data: Optional: information in this parameter supplied by p-user is conveyed as a transparent binary octet string by the p-service. Maximum length is 48 octets.

Usage and Effect

The request primitive may only be issued by the current owner of the negotiate-token. Effects are sequenced and non-destructive.
In the Draft PE the parameters referenced in p-param-list change state to "agreed/negotiable". Some other parameters may revert from "frozen" to "agreed/negotiable" when the negotiate-token is next passed, or when the negotiation-enclosure is successfully terminated with option "switch", whichever happens first.

2.2.8 Information Handling Facilities and Transfer Enclosure

The principal reason for the establishment of the p-connection and the various parameters thereof is to enable the p-users to exchange or manipulate information of mutual interest. This activity is performed within the context of a Defined Presentation Environment (PE).

Details of the information handling requirements and facilities are necessarily very dependent on the nature of the service required by the p-users as established at connection establishment and PE negotiation.

The following material thus gives only general aspects of this part of the Presentation Service. Each individual Service and Class Standard will need to build on this general information with the detail applicable to its methods of operation. Those aspects which are defined herein in some detail are expected to be applicable to all or at least most services and classes. Each Standard will indicate applicability of any parts of this general information and give any variations and further information as necessary to complete the definition in that Standard.

The number of presentation environments which can exist concurrently during a p-connection determines the requirements for multiple-pe option.

2.2.8.1 The Transfer Enclosure

This is the formal concept within which a presentation environment (PE) is used to enable information handling to occur according to the definition of the PE. Information handling is possible within transfer-enclosure only if a Defined PE has been assigned, either implicitly or explicitly.

Entry to the Transfer Enclosure will be implicit following successful Establishment by P-CONNECT if a Defined

PE then exists according to the parameters of P-CONNECT. Explicit entry can be caused by use of P-CHANGE-ENCLOSURE service element.

The PE in use in transfer-enclosure is known as the Current PE. If the negotiation facility is available in the current subset the single-interaction negotiation (P-PERFORM-NEGOTIATION) can be used to alter the parameters of the Current PE.

Termination of transfer-enclosure can be by explicit confirmed service element (P-CHANGE-ENCLOSURE), and in this case the current results of the information handling can be retained for later use (subject to service dependent restrictions), can be unilaterally forced by either p-user, or is caused implicitly if P-RELEASE (successful), P-DISCONNECT or P-ABORT occurs; in these cases the results are lost (from the scope of the p-service).

The right to initiate transfer-enclosure is subject to the rules for the use of P-CHANGE-ENCLOSURE. Activities within transfer-enclosure are generally mediated by the transfer-token, initially held on entry by the issuer of the P-CHANGE-ENCLOSURE request primitive. Some classes and PEs may use additional tokens or make use of the transfer-token for additional functions.

2.2.8.2 Multiple-Presentation Environment Option

If more than one concurrent Defined PE is required, the Multiple-PE Option must be agreed at establishment time by P-CONNECT. Selection of the PE to be used, from any PEs which are available in a fully Defined state, is made at entry to transfer-enclosure and cannot be changed within transfer-enclosure.

Negotiation facilities are used to set up Defined PEs required in addition to the one, the "initial" PE, which can be set up at establishment time; both single- and multiple- interaction forms of negotiation can be used for this purpose.

Defined PEs additional to "initial" are identified by "pe-ident/n" values given numeric identifiers from "1" upwards.

2.2.8.3 Entry to Transfer Enclosure

The conditions of entry to the transfer-enclosure, which will determine the overall success of the P-CHANGE-ENCLOSURE service element as described in 2.2.4, are given by the following enclosure specific information:

p-dest-encl-type: takes value "transfer".
p-entry-params: as table and descriptions below.

Entry parameters for Transfer-enclosure:

Parameter Name	req	ind	resp	conf
p-pe-ident	D	U	x	x
p-start-pe-option	D	U	x	x

Parameter descriptions

p-pe-ident: Identifies the PE which is to be used in the transfer-enclosure. Optional, and Invalid if multiple PE option is not in use. Symbolic values are "initial" which selects the PE defined at establishment time, or "pe-ident/n" value which selects the designated PE. The designated PE must be a Defined PE. Default value if parameter is not present is "initial".

p-start-pe-option: Optional: this can be used to select an initial state on entry to transfer-enclosure: symbolic value "initial" (the default) will initialise the PE according to its definition, "resume" will cause the PE to adopt the state and content at the last "save" if any. Provision of "resume" capability and hence applicability of this parameter is service and class dependent; parameter is invalid if p-save-pe-capab in P-CONNECT is not "yes".

Usage and Effects

See description of P-CHANGE-ENCLOSURE for general information on usage and effects. Entry will not be successful if the designated PE is not in Defined state or if p-start-pe-option requests "resume" and this capability is not available. Entry to transfer-enclosure will be successful but a "success with warning" diagnostic will be generated if "resume" is requested but there is no "saved" information for the PE; the initial state of the PE will be taken.

Additional information

The transfer-token will be held initially by the p-user which issued the (successful) request primitive. The initial allocation of any other tokens for use in transfer-enclosure is service and class dependent and will be given in individual standards. A later version of this Standard may include extension of transfer-enclosure exit and entry parameters to provide a check-point and restart capability.

2.2.8.4 Exit From Transfer Enclosure

Transfer-enclosure can be terminated with the agreement of both p-users by means of P-CHANGE-ENCLOSURE or, subject to service specific restriction, forced unilaterally by either of them by means of P-CLOSE/ ABORT-ENCLOSURE.

2.2.8.4.1 Agreed Exit

This uses P-CHANGE-ENCLOSURE with exit parameters, which condition the success of the service element as described in 2.2.4, as described below:

p-exit-params: As table and descriptions below.

Exit parameters from Transfer-enclosure:

Parameter Name	req	ind	resp	conf
p-save-pe-option	D	U	x	x

Parameter descriptions

p-save-pe-option: Optional: allows the request issuer to indicate what should be done with any information stored in the p-service relating to the Current PE if the enclosure is ended; symbolic value "save" will store the state and content for possible restart, "discard" will discard this information (from the scope of the p-service).

The availability of save/resume facility is service and class dependent. If parameter is omitted "discard" always occurs. If parameter is present (either value) but save capability is not available the service element will be rejected with appropriate p-diagnostic.

Usage and Effects

See description of P-CHANGE-ENCLOSURE for general information on usage and effects.

Additional information

The current information of the transfer-enclosure, if retained as a result of "save" value of p-save-pe-option, remains associated with the pe-ident used in the transfer-enclosure. It will be lost if the same pe-ident is selected for re-use in transfer-enclosure unless the p-start-pe-option value is "resume". See also Additional information under entry to transfer-enclosure.

2.2.8.4.2 Unilateral Close/Abort

This uses P-CLOSE/ABORT-ENCLOSURE; there are no exit conditions as the service element is always successful, and the null-enclosure is always entered after the completion of the service element.

Enclosure-specific information

When transfer-enclosure is aborted the effects of service elements of the transfer-enclosure which were in progress will be lost. After completion of this service element the enclosure-token, if defined, is owned by the initiator of the aborted transfer-enclosure. The status of the state and content of the PE after this service element depends on the checkpointing facilities offered by the specific service. A later version of this Standard may include further general facilities.

2.2.8.5 Information Handling Services

The exact nature of the service elements required for information handling within transfer-enclosure for different presentation services necessarily vary quite widely. At one extreme there can be merely a simple information-passing facility with simple serial syntax adjustments and no p-service control over dialogue, at the other there may be the maintenance in the p-service of a complex data structure with multiple access pointers.

For this reason the Generic Presentation Service does not contain any specific services for use within transfer-enclosure and specific service standards will include the necessary services to meet the individual requirements. In some cases these may include a more or less direct visibility to the p-users of some parts of the session service according to the session facilities selected at establishment time.

2.2.9 Service Exception Conditions

Exception conditions visible in the Service Description are either p-service detected or p-user advised.

Some services and classes of p-service are likely to provide capabilities for recovering from exceptions of both types as above, similar to or extended forms of the recovery facilities available to p-service from session layer.

Currently, the Generic Presentation Service Description includes only the P-ABORT as available to the p-service in the event of an exception condition in the p-service itself (invalid protocol, mis-use by a p-user, etc.), and P-DISCONNECT as available to a p-user needing to abort the p-connection for any reason.

2.2.10 Diagnostic Information

All Confirmed service elements provide a p-diagnostic parameter whereby the success or otherwise of the service element can be advised to the p-user(s).

The p-diagnostic parameter conveys up to three items of information corresponding to three levels of analysis:

- severity
- reason
- diagnostic supplement

Each element can be included only if the preceding (more synthetic) elements have been supplied. A separate diagnostic parameter is used for each detected error. Limitations specific to some services are indicated within the specific descriptions of these services in earlier sub-clauses.

2.2.10.1 Severity

Specifies the degree of success or failure. The legal values, by increasing order of severity:

- "success"
- "success with warning"
- "failure"

Provision of severity is mandatory.

2.2.10.2 Reason

Gives a summary diagnostic. It is not used when severity is "success". The lists of legal values applicable to each service are currently not given at the level of the service description but will be found in the definitions of the protocol messages and parameters in Section 3. The reasons for failure of a service element are generally described qualitatively in the service description in this Section.

2.2.10.3 Diagnostic Supplement

Gives more detailed information on the error event. There can be several diagnostic supplements in one diagnostic parameter. Three types of diagnostic supplement are defined as follows.

DS-0:

May accompany any reason value and is specifically expected when the reason value is "non-standard-reason". The legal value is a character-string of a maximum length of 31 characters.

DS-1:

Accompanies standard reason values associated with incorrect parameter values. Repeated for each erroneous parameter and contains the type of the defective parameter.

DS-2:

Accompanies standard reason values associated with protocol violation and contains the type of the erroneous message.

3 Protocol

3. PROTOCOL

3.1 Protocol Overview

The communication between the two General Presentation Service users is effected by means of a Presentation Protocol operated by the presentation-entities and defined in terms of Protocol Messages and Protocol Sequences. This Section of the Standard defines general aspects as a framework for detailed definitions in service and class standards.

The functions offered by the protocol are not in general directly accessible to the service users (i.e. the p-users) but are employed by the presentation entities as required to support the services made available to the service user by means of a local service interface. There will not necessarily be a one-for-one correspondence between interface primitives and protocol messages or actions. The Service standards define the protocol messages and rules for use, they do not define the local service interfaces. However, in some cases there will be an essentially one-for-one mapping for many aspects of the p-service.

3.1.1 Protocol Messages

A protocol message is the smallest quantity of information exchanged between peer entities which has a self-contained semantic significance. Although it is recognised that a message may have an internal structure (for instance it may imply the use of more than one Session Service primitive), for the purpose of communication between p-entities a message is considered to be an indivisible unit.

Request and response types of message are recognised. Some messages occur in isolation and do not require a particular message as answer. Such messages are called request messages and each is said to be the (only) member of a type 1 protocol message group. Other messages require the receiving p-entity to respond with a particular message (sometimes the response must be generated immediately, sometimes other messages may be transmitted before the response is given). The first message of such a pair is a request message, the second is a response message. The message pair is referred to as a type 2 protocol message group.

A protocol message contains protocol control information (i.e. header and possibly one or more parameters) and may contain data.

In order to define the protocol it is necessary to define under which conditions (i.e. in which states) a certain message may be sent to the peer entity and which messages should be accepted from the peer entity. In all cases the actions and resulting state must be specified.

3.1.2 Summary List of Protocol Messages

PROTOCOL MESSAGES		MINIMUM SUBSET
CONNECT protocol messages		
PP-CONNECT-REQ	CNQ	
PP-CONNECT-RESP	CNR	
RELEASE protocol messages		
PP-RELEASE-REQ	RLQ	
PP-RELEASE-RESP	RLR	
DISCONNECT protocol message		
PP-DISC	DNQ	
CHANGE-ENCLOSURE protocol messages		
PP-CHANGE-ENCL-REQ	CEQ	C *
PP-CHANGE-ENCL-RESP	CER	C *
CLOSE/ABORT-ENCL protocol messages		
PP-ABORT-ENCL-REQ	AEQ	C
PP-ABORT-ENCL-RESP	AER	C
PERFORM NEGOTIATION protocol messages		
PP-PERFORM-NEG-REQ	PNQ	B
PP-PERFORM-NEG-RESP	PNR	B
NEGOTIATION ENCLOSURE protocol messages		
PP-NEG-INVITE	NIQ	C
PP-NEG-OFFER	NOQ	C
PP-NEG-ACCEPT	NAQ	C
PP-NEG-REJECT	NRQ	C

Legend

Minimum Subset: Indicates the minimum GPP subset at which the message is applicable. No entry implies that the message is applicable in all GPP subsets.

* : restricted parameters at subset C.

No other protocol messages are defined by GPP itself.

3.1.3 Conventions and Notation for Definition of Protocol Messages

Clause 3.2 provides a narrative definition of the protocol messages with indication of the protocol message groups in which they are used.

Each message is defined by the following items:

- purpose, with associated message, if any,
- list of parameters,
- brief description of generation and reception conditions.

A detailed definition of the parameters is given only when they differ from the description of equivalent parameters of the service elements in Section 2.

3.1.4 Mapping of Messages into Lower Layer Service

This mapping is not considered in detail in the narrative description in Clause 3.2 but is taken into account in Clause 3.3 which defines the encoding of the presentation protocol messages. Definition of the mapping is given in Clause 3.4. A consequence of the mapping is that some of the parameters in the messages will not need an explicit encoding in the Presentation Protocol Encodings since they are sent using specific parameters of the lower (session) layer services other than SS-user-data.

3.2 Protocol Message Definitions

This Clause defines the Protocol Messages which are generally applicable to Presentation Protocols. It also gives the guidelines which are followed by individual service and class standards for the definition of elements of protocol and parameters which are service and class dependent. General aspects of Encoding are defined in Clause 3.3 and of the mapping of elements of presentation protocol into Session Service primitives in Clause 3.4.

Individual Standards will state the extent to which the general definitions apply and give any additional details which are needed to complete the specification of the protocol for that Standard.

The protocol messages are defined for each protocol structure. The messages are given full names which should be used outside the context of this Standard, but abbreviated names are given which are used elsewhere in this Standard for brevity.

3.2.1 CONNECT Protocol Messages

PP-CONNECT-REQ (CNQ)

Purpose:

To request the establishment of a presentation-connection and propose parameter values; solicits a CNR response.

Contents:

Parameters derived from the parameters shown as relevant to P-CONNECT request primitive as follows:

pp-protocol-definition: Derived from service parameters p-service-ident, p-service-class-ident, p-service/class-version-idents, p-service/class-subset-ident, p-pe-profile.

pp-session-subset: Derived as for pp-protocol-definition.

pp-session-parameters: Derived from pp-protocol-definition, and possibly from other parameters of P-CONNECT; see Additional information.

pp-user-caller:
pp-user-called:
pp-fixed-pe-params:
pp-initial-pe-params:
pp-max-pes:
pp-save-pe-capab:
pp-transp-data:
pp-special-conventions:

These parameters are derived directly from the service parameters excluding any element mapped into the above protocol parameters.

Some parameters may not be present.
See also Additional Information.

Generation

Generated when a p-user issues a valid P-CONNECT request primitive.

Reception

A valid CNQ will cause a P-CONNECT indication primitive from the PPM. If the p-service finds the parameters acceptable this indication will be passed to the p-user designated in the p-user identification information; otherwise the p-service will itself issue a "reject" response primitive which will result in generation of a CNR protocol message with appropriate diagnostic.

Additional information

The details of the mapping between the service parameters and the protocol parameters are specific service dependent and will be given in the individual service standards. With this protocol version session service parameters cannot be contained within service parameter p-initial-pe-params.

Parameter p-protocol-definition is implicitly part of p-fixed-pe-params if it appears separately. It can, however, appear as a component parameter of p-fixed-pe-params, in which case its effects are "fixed", and/or as a component parameter of p-initial-pe-params in which case its effects are not fixed; see 3.3.6.1.

PP-CONNECT-RESP (CNR)

Purpose:

Response to CNQ to indicate success or failure of connection attempt.

Contents:

Parameters derived from the parameters shown as relevant to P-CONNECT response primitive as follows:

pp-protocol-definition: derived from service parameters p-service/class-versions, p-service/class-subset-ident, p-pe-profile.

pp-session-params: derived as for pp-protocol-definition and, possibly, from other parameters of P-CONNECT.

pp-user-called:
pp-fixed-pe-params:
pp-initial-pe-params:
pp-max-pes:
pp-save-pe-capab:
pp-transp-data:
pp-special-conventions:

Derived directly from the service parameters with exclusion of elements mapped into the above protocol parameters.

pp-diagnostic: derived from the service parameter p-diagnostic or supplied by the p-entity.
Some parameters may not be present, pp-diagnostic may be omitted if severity value is "success".

Generation

Generated by p-entity when the p-user has issued a P-CONNECT response primitive or when the p-entity is itself refusing to accept the connection.

Reception

A valid CNR will result in a P-CONNECT confirmation primitive to the p-user with appropriate p-diagnostic parameter.

Additional information

If the diagnostic is present and the severity value indicates other than "success" or "success with warning" the connection has not been established and the p-entities return to their previous state. The meaning of "success with warning" is service dependent and this value may not be used in some cases. See also Additional information to CNQ.

3.2.2 RELEASE protocol messages

PP-RELEASE-REQ (RLQ)

Purpose:

To request the tidy termination of a p-connection; solicits a RLR response.

Contents:

Parameters derived from the parameters shown as relevant to P-RELEASE request primitive as follows:
pp-transp-data: derived directly from p-transp-data.

Generation

Generated if p-user issues P-RELEASE request primitive. Is sent following any outstanding protocol messages for protocol message groups initiated by the p-entity or needing completion by the p-entity.

Reception

A valid RLQ element of protocol will result in a P-RELEASE indication primitive to the p-user.

PP-RELEASE-RESP (RLR)

Purpose:

Response to RLQ to indicate success or failure of release attempt.

Contents:

Parameters derived from the parameters shown as relevant to P-RELEASE response primitive as follows:
pp-transp-data: derived directly from p-transp-data,
pp-diagnostic: derived from the service parameter p-diagnostic or supplied by the p-entity.
Some parameters may not be present; pp-diagnostic may be omitted if severity value is "success".

Generation

Generated by p-entity when the p-user issues a P-RELEASE response primitive. Is sent following any outstanding protocol messages for protocol message groups initiated by the p-entity or needing completion by the p-entity.

Reception

A valid RLR results in a P-RELEASE confirmation primitive to the p-user with appropriate p-diagnostic.

Additional information

Validity of the failure case depends on the session subset in use, see sub-clause 3.4.4. "failure" is valid only if a token has been defined for mediating RELEASE. In the case of (valid) failure to accept Release, the p-connection is retained and the p-entities adopt the states before the issue of the request/indication primitives.

3.2.3 DISCONNECT protocol message

PP-DISC (DNQ)

Purpose:

Used to force disconnection of the p-connection and the supporting s-connection when requested by a p-user or when found necessary by a p-entity. No response is solicited.

Contents:

Parameter pp-disc-reason containing either the p-transp-data from the P-DISCONNECT request primitive or a diagnostic code giving the p-entity's reason for forcing the disconnection.

Generation

Generated by p-entity when p-user issues P-DISCONNECT request primitive, or when a p-entity detects an exception condition which forces abort of the p-connection.

Reception

A DNQ will cause the p-entity to abandon any activities in progress or queued and give P-DISCONNECT indication or P-ABORT indication primitive to the p-user according to the parameter value.

Additional information

Internal p-service abort and abort due to S-ABORT indication(s) are notified to the p-user(s) outside the scope of this Standard.

3.2.4 CHANGE ENCLOSURE protocol messages

PP-CHANGE-ENCL-REQ (CEQ)

Purpose:

To request an orderly change of enclosure from the current one to a designated one; solicits a CER response.

Contents:

The parameters below are derived directly from the parameters shown as relevant to P-CHANGE-ENCLOSURE request primitive:
pp-dest-encl-type, pp-exit-params, pp-entry-params, pp-transp-data.
Some parameters may not be present.

Generation

When the p-user issues a P-CHANGE-ENCLOSURE request primitive with parameters acceptable to the associated p-entity.

Reception

A valid CEQ will cause a P-CHANGE-ENCLOSURE indication primitive to the p-user. The parameters of the indication primitive may have been changed by the receiving presentation entity to reflect its capabilities. If the p-entity is unable to execute the request a CER will be generated without indication to the p-user.

PP-CHANGE-ENCL-RESP (CER)

Purpose:

Response to CEQ to indicate success, partial success or failure of a request to make an orderly change from one enclosure to another.

Contents

The parameters below are derived directly from the parameters shown as relevant to P-CHANGE-ENCLOSURE response primitive:
pp-dest-encl-type, pp-transp-data; pp-diagnostic.
Some parameters may not be present; pp-diagnostic may be omitted if the severity value is "success".

Generation

When the p-entity itself rejects the CEQ or when the p-user issues a P-CHANGE-ENCLOSURE response primitive.

Reception

A valid CER causes a P-CHANGE-ENCLOSURE confirmation primitive to the p-user with appropriate p-diagnostic parameter.

3.2.5 CLOSE/ABORT-ENCLOSURE protocol structure

PP-ABORT-ENCL-REQ (AEQ)

Purpose:

To request a forced close/abort of the current enclosure and entry to null-enclosure; solicits an AER response.

Contents:

The parameter below is derived directly from the parameter shown as relevant to P-CLOSE/ABORT-ENCLOSURE request primitive:
pp-transp-data.
This parameter may not be present.

Generation

When the p-user issues a P-CLOSE/ABORT-ENCLOSURE request primitive.

Reception

An AEQ will cause the p-entity to abandon any activities in the current enclosure, close the enclosure and give a P-CLOSE/ABORT-ENCLOSURE indication primitive to the p-user. Null-enclosure is entered. For effects on the closed enclosure, see service element descriptions.

PP-ABORT-ENCL-RESP (AER)

Purpose:

To confirm the forced enclosure change requested by an AEQ message.

Contents:

The parameter below is derived directly from the parameter shown as relevant to P-CLOSE/ABORT-ENCLOSURE response primitive:
pp-transp-data.
This parameter may not be present.

Generation

When the p-user issues a P-CLOSE/ABORT-ENCLOSURE response primitive.

Reception

A valid AER will cause the p-entity to give a P-CLOSE/ ABORT-ENCLOSURE confirmation primitive to the p-user. Null-enclosure is then entered. For effects on the closed enclosure, see service element descriptions.

3.2.6 PERFORM NEGOTIATION protocol messages

PP-PERFORM-NEG-REQ (PNQ)

Purpose:

To request the performance of a single-interaction negotiation enclosure; solicits a PNR response.

Contents:

The parameters below are derived directly from the parameters shown as relevant to P-PERFORM-NEGOTIATION request primitive:
pp-pe-ident; pp-draft-pe; pp-pe-specific-params; pp-special-info, pp-transp-data.
Some parameters may not be present. See Additional information.

Generation

When the p-user issues a P-PERFORM-NEGOTIATION request primitive with parameters acceptable to the associated p-entity.

Reception

A PNQ causes a P-PERFORM-NEGOTIATION indication primitive to the p-user unless the p-entity is unable to execute the request in which case a PNR is generated.

Additional information

Parameter pp-pe-specific-params may occur more than once and specific service standards may define more than one form of this parameter.

PP-PERFORM-NEG-RESP (PNR)

Purpose:

Response to PNQ to indicate success or failure of the negotiation attempt.

Contents:

The parameters below are derived directly from the parameters shown as relevant to P-PERFORM-NEGOTIATION response primitive:

pp-pe-specific-params; pp-special-info,
pp-transp-data; pp-diagnostic.

Some parameters may not be present; pp-diagnostic may be omitted if the severity value is "success". See also Additional information.

Generation

When the p-entity rejects a CNQ negotiation request or when the p-user issues a P-PERFORM-NEGOTIATION response primitive.

Reception

A PNR causes a P-PERFORM-NEGOTIATION confirmation primitive to the p-user.

Additional information

If the diagnostic severity indicates other than "success" there is no overall result of the message group. Severity value "success with warning" is not relevant. See also Additional information to PNQ.

3.2.7 NEGOTIATION ENCLOSURE protocol messages

PP-NEG-INVITE (NIQ)

Purpose:

To request an offer for a set of parameters. No response is explicitly solicited as part of this protocol message group.

Contents:

The parameters below are derived directly from the parameters shown as relevant to P-NEG-INVITE request primitive:

pp-param-ident-list; pp-transp-data.

Some parameters may not be present.

Generation

When the p-user issues a P-NEG-INVITE request primitive with p-invited-param-list compatible with the current state of the Draft PE.

Reception

A valid NIQ causes a P-NEG-INVITE indication primitive to the p-user.

PP-NEG-OFFER (NOQ)

Purpose:

To offer one or more values or ranges of values for a set of parameters. No response is explicitly solicited as part of this protocol message group.

Contents

The parameters below are derived directly from the parameters shown as relevant to P-NEG-OFFER request primitive:

pp-param-list; pp-transp-data.

Some parameters may not be present.

Generation

When the p-user issues a P-NEG-OFFER request primitive with p-offer-param-list compatible with the current state of the Draft PE.

Reception

A valid NOQ causes a P-NEG-OFFER indication primitive.

PP-NEG-ACCEPT (NAQ)

Purpose:

To indicate acceptance of the value or value range of a set of parameters. No response is explicitly solicited as part of this protocol message group.

Contents:

The parameters below are derived directly from the parameters shown as relevant to P-NEG-ACCEPT request primitive:

pp-param-list; pp-transp-data.

Some parameters may not be present.

Generation

When the p-user issues a P-NEG-ACCEPT request primitive with p-accept-param-list compatible with the current state of the Draft PE.

Reception

A valid NAQ element of protocol causes a P-NEG-ACCEPT indication primitive to the p-user.

PP-NEG-REJECT (NRQ)

Purpose:

To indicate rejection of one or more parameters in a previous offer. No response is explicitly solicited as part of this protocol message group.

Contents:

The parameters below are derived directly from the parameters shown as relevant to P-NEG-REJECT request primitive:

pp-param-ident-list; pp-transp-data.

Some parameters may not be present.

Generation

When the p-user issues a P-NEG-REJECT request primitive.

Reception

A valid NRQ element of protocol causes a P-NEG-REJECT indication primitive to the p-user.

3.2.8 Information Handling Facilities

The generic protocol definition does not include definition of specific protocol messages or parameter content for information handling as these are specific-service-dependent. Clause 3.3 reserves ranges of coding values for use for such messages.

3.3 Protocol Encoding

Bit numbering convention: the bits of an octet are identified as f1, f2, f3, ..., f7, f8, where f1 is the left-most and most-significant bit of the octet. A similar convention applies to bit strings longer than one octet, with the right-most and least-significant bit becoming f16, f24, etc.

Note 3

'f' is used in this convention to distinguish this convention from that used in character coding tables, etc., where the numbering is in the reverse direction, ie. b1 is the least-significant bit, and extension, e.g. from 7-bit to 8-bit codes, is by addition to the left.

3.3.1 Protocol Message Structure

The protocol messages defined in clause 3.2 are transferred between the peer p-entities using the services of the session connection in different ways.

Each protocol message is composed of two parts:

- a one octet presentation-header,
- a variable-length presentation-message-content.

Some messages are uniquely mapped into a session-service primitive, i.e. form the only possible content of such primitive; the presentation header can be omitted since this is implied by the session-service used for the transfer of the p-message. Whether this is actually done is defined in clause 3.4. Frequently for such messages some part of the content as defined in clause 3.2 will be mapped into the specific parameters of the session service primitive and such parts will, in general, not be explicitly encoded in the presentation protocol data unit which is mapped into the SS-user-data field (if any) of the primitive.

Some other messages are completely mapped, including both header and all content, into the SS-user-data field of S-DATA service.

A remaining category of message requires more than one session service primitive, when, in addition to the passing of data, there is an implicit requirement to initiate a specialised session service, for example the passing of a token.

Full definition of the mapping between presentation protocol messages and the session service primitives is given in clause 3.4.

3.3.2 Presentation Header

The single-octet presentation header is used as an 8-bit Message Type with values 0 to 255 encoded in binary.

3.3.2.1 Principle of Allocation of Message Type

The value range is divided up as follows:

- 0: Reserved, not used as a Type Code.
- 1 to 63: Allocated for presentation protocol control messages defined within the Generic Presentation Protocol Standard. Values are assigned from 1 upwards.
- 64 to 255: Allocated as available for definition in specific protocol standards without requirement for uniqueness between such standards.

3.3.3 Presentation Message Content

The structure of this is dependent on the message type. In many cases it will consist of one or more parameters encoded as defined in 3.3.4; in other cases there will be an implicit encoding of the information, dependent on the message type and defined for each case. The final (unstructured) part of the content may not have an explicitly encoded length but be implicitly bounded by the SSDU into which the message is mapped.

3.3.4 Parameter Encoding

3.3.4.1 Type/Length/Value (TLV) Encoding

The TLV technique is a method for coding of an information unit such as a "parameter".

As used in this Standard each such unit is made up of three parts:

- type (first field)
- length (second field)
- value (third field)

Type field - 1 octet

f1 = 0
f2-f8 = type value: 1 to 127, 0 reserved.

Length field - 1 or 2 octets

f1	= 0 :	length L specified on 7 bits (f2-f8)
	= 1 :	length L specified on 15 bits (f2-f16)
f2-f8/f16	:	binary number L of octets of the value field.

Value field - variable length

L octets of data, internal structure and coding is dependent on Type, and may be a nested TLV (compound value).

3.3.4.2 Parameter Value Representation

When specifying the type of encoding used in the value field of parameters the following convention is used:

C: character string: Any size (unless limits are explicitly defined). Character encoding is according to the ECMA 7-bit coded character set (see ECMA-6). Each character is mapped on bits f2-f8 of an octet, with bit f1 = 0. The allowed characters are those in columns 2 to 7 of the International Reference Version, excluding that in position 7/15. Particular parameters may apply further constraints on the range of characters.

N: numeric: Unsigned binary integer with four possible sizes: 8, 16, 24, or 32 bits (L=1, 2, 3 or 4).

S: symbolic: Unsigned binary integer the values of which encode specific meanings. Size is 8 bits (L=1).

M: bit-map: Bit string in which each bit encodes a specific meaning. Size is 8 or 16 bits (L=1 or 2). Any bit the role of which is not defined must be encoded as 0. For bits representing specified options, the bit is set to 1 if the option is requested, to 0 otherwise.

T: transparent: Coding and meanings explicitly defined or p-service is transparent to it. Conveyed as a binary octet string.

A: aggregate value: Value is a structure of fields explicitly defined for the particular parameter; the fields may but will not necessarily conform to the above classifications, in particular the sizes may be smaller. Conveyed as a binary octet string.

P: compound parameter: Value is made up of a nested structure of parameters, see 3.3.4.4.

Value truncation: For economy reasons it is recommended to use the minimum variable length string for expressing values:

- character strings should not contain unnecessary spaces,
- for numeric values, all unnecessary octets from left containing only zero bits should be removed,
- for symbolic values, the octet may be removed (length becomes 0) if the symbolic value is that corresponding to binary value 0,
- for bit-map values, all unnecessary octets from right containing only zero bits should be removed (extension of bit-map values will be to the right, thus the length may become 0),
- for transparent values, no specific truncation recommendation can be given; particular definitions may define a truncation algorithm.

3.3.4.3 Parameter Encoding

For each parameter of the protocol, the following information is given:

- parameter type value: this is unique among the parameters which can occur at the particular level of the "parameter tree",
- value representation type: C, N, S, M, T, A,or P.
- maximum length of value field: "-" means unlimited (in the general definition),
- encoding of all possible values: for S fields,
- significance of all bits: for M fields,
- all possible component parameters: for P values,
- detailed structure: for A values.

For parameters which are mapped directly into the parameters of the session service this information is not usually required where the relevant information is given in the session standard, but supplementary information may be needed and will be given in clause 3.4.

Default Values: The existence of a default value which will be assumed if the parameter is not explicitly specified, depends partly on the value type for the parameter and partly on the specific definition of the parameter:

- for value types S and M there will generally be a default value, equal to 0,
- for value types C and T there is no default,
- for value type N there is generally no default value, but a default may be defined for some such parameters.

3.3.4.4 Nested Parameters

In general the value field of a TLV-encoded parameter can itself be a collection of TLV-encoded parameters. Such a parameter is known as a compound parameter, and its value as a compound value; value type designation is "P". Each of the contained parameters can also be a compound parameter, to any required depth of nesting. Where a parameter value is not one or more TLV parameters it is a simple parameter of the nested parameter structure. The value of such a parameter can be either an elementary value (value types C, N, S, M, T) or an aggregate value (value type A); in the "A" case there is an explicit substructure defined explicitly for that parameter.

This Standard includes a number of compound parameters, usually without detailed definition of the contents which are generally specific service dependent. Parameters specific to a service which are always contained within a particular "outer" parameter need to have type values unique among the total set of parameters which can occur within the value field of the outer parameter but which can overlap the type values of other parameters which can never occur at this "node".

3.3.4.5 Range and List Parameters

To enable the declaration, in connection with a particular parameter type, of a range of values, i.e. with a low limit and a high limit, or of a list of specific discrete values, two special parameter type codes are defined as follows.

Range Parameter: type 32

List Parameter: type 31

These always have a compound value. The components can be a further Range or List parameter or can be some other parameter type; in this second case these component parameters must be of the same type.

In the case of a Range parameter the components will be two in number; the first gives the low limit and the second the high limit.

In the case of a List parameter any number of components can be present, each giving a specific entry value for the list; this value can be a range, a list, or a single compound or simple parameter.

3.3.5 Message Type Codes

The following table gives the allocation of Message Type Codes in the Presentation Header for the protocol messages defined in this Standard.

PROTOCOL MESSAGE		TYPE CODE
PP-CONNECT-REQ	CNQ	1
PP-CONNECT-RESP	CNR	2
PP-RELEASE-REQ	RLQ	3
PP-RELEASE-RESP	RLR	4
PP-DISC	DNQ	(5) *
PP-PERFORM-NEG-REQ	PNQ	8
PP-PERFORM-NEG-RESP	PNR	9
PP-CHANGE-ENCL-REQ	CEQ	10
PP-CHANGE-ENCL-RESP	CER	11
PP-ABORT-ENCL-REQ	AEQ	12
PP-ABORT-ENCL-RESP	AER	13
PP-NEG-INVITE	NIQ	24
PP-NEG-OFFER	NOQ	25
PP-NEG-ACCEPT	NAQ	26
PP-NEG-REJECT	NRQ	27

* This type value is reserved but is not used since the message is mapped into a session service primitive.

3.3.6 Parameter Encoding Details

The following table lists the parameters which are standardized in this Standard and which conform to the general principles in 3.3.4. Some special cases are covered in 3.3.7. Numbers in parentheses refer to additional information following the tables.

The rule for ordering of parameters encoded within protocol messages or within compound parameters is implied by the order in which the parameters are given in parameter lists or tables, unless some other specific statement is made. A receiving p-entity can, but is not obliged to, consider as invalid a message which violates such a rule.

GPP Parameters

PARAMETER NAME	TYPE	LENGTH (MAX)	VALUE TYPE	VALUE ENCODING
pp-diagnostic	1	-	A	aggregate value, see 3.3.7.1.
pp-transp-data	2	48(1)	T	
pp-protocol-definition*	3	-	A	aggregate value, see 3.3.7.2.
pp-fixed-pe-params	4	3(2)	P	compound value (service specific).
pp-initial-pe-params	5	3(2)	P	compound value (service specific).
pp-max-pes	6	2	N	bin value: maximum number of pes; 0,1 are invalid.
pp-pe-ident	7	2	N	bin val 0:"initial" 1 upwards: pe-ident from numeric list.
pp-draft-pe	8	2	N	bin val 0:"initial" 1 upwards: pe-ident from numeric list.
pp-dest-encl-type	9	1	S	0: not used, 1:"null", 2:"transfer" 3:"negotiation"; other values reserved.
pp-pe-specific-params	10 (3)	-	P	compound value, service specific.
pp-save-pe-capab	11	1	S	0: not used, 1:"yes", 2: "no"; other values reserved.
pp-pe-action	15	1	S	0: not used 1:"switch", 2:"restore", 3:"delete"; other values reserved.
pp-param-ident-list	16	-	P	compound value, service specific.
pp-param-list	17	-	P	compound value, service specific.

PARAMETER NAME	TYPE	LENGTH (MAX)	VALUE TYPE	VALUE ENCODING
pp-start-pe-option	20	1	S	0: "initial", 1: "resume"; other values reserved.
pp-save-pe-option	21	1	S	0: "discard", 1: "save"; other values reserved
pp-special-conventions	22	2	T	
pp-special-info	23	-	T	

* : See Additional information in sub-clause 3.3.6.1

Additional information

1. This maximum length may be explicitly reduced when this parameter is used in particular protocol structures; the maxima applicable are given in the service element descriptions to which the protocol message definitions refer.
2. This maximum length may be increased subject to trade-off with other parameters; this is explicitly stated in the service element description.
3. In addition to this Standard compound parameter type 10, parameter type codes 64 to 127 are reserved for specific forms of pp-pe-specific-params which are expected to be simple parameters but may have aggregate values.

3.3.6.1 Additional Information on Use of pp-protocol-definition parameter

This parameter, as defined in sub-clause 3.3.7.2, can have from 3 to 5 fields. The fields which are present may contain "non-void" values or can be "void".

a) pp-protocol-definition can be encoded at the outermost level of parameter encoding in the CNQ and CNR messages. (It will be the first parameter in the message). In this case any "non-void" contents of it are deemed to be part of pp-fixed-pe-params; this parameter itself may or may not be explicitly present with some other contents.

b) pp-protocol-definition can be encoded as the first component parameter of pp-fixed-pe-params; the interpretation of any "non-void" contents is the same as in the above case.

It is not likely, but is not excluded, that pp-protocol-definition will appear in both of the above ways in any one protocol message. If it does its contents must not conflict.

c) pp-protocol-definition can be encoded as the first component parameter of pp-initial-pe-params; in this case any "non-void" contents are interpreted with the same status as other parts of this parameter i.e. are not "fixed". The contents must not conflict with any useful contents encoded in either of the above two ways.

3.3.7 Special Parameter Encodings

In some cases, because of limited availability of space for a parameter, e.g. due to a restriction in the session service parameters, or the desire to keep the overheads to a minimum for a frequently used message, the parameters are not encoded using the standard scheme as defined in 3.3.4.

3.3.7.1 Encoding of Diagnostic Parameter

In most messages this is encoded using a standard TLV format and the value of Type is given in the table above. The value is an "aggregate" parameter as given below.

The value field for pp-diagnostic is a structure defined as follows using BNF notation:

<diagnostic value>::=<severity><reason> [<DS>]

[ ] indicates that this element may be repeated.

<DS> ::= <DS type><DS value length><DS value>
The terminal elements are encoded as follows:

<severity>
4-bit binary integer; f1-f4 of first octet of <diagnostic value>. Values are defined below. The default value assumed when the diagnostic parameter is omitted is 0 (success).

<reason>
12-bit binary integer: f5-f8 of first octet of <diagnostic value> followed by f1-f8 of second octet. Values are defined below together with the messages in which the values can appear. The default value assumed when the diagnostic parameter is omitted is 0 (no reason given).

<DS type>
3-bit binary integer; f1-f3 of first octet of <DS>. Values 0,1,2 are used, other values are reserved.

<DS value length>
5-bit binary integer; f4-f8 of first octet of <DS>.

<DS value>
Dependent on <DS type> as follows:

DS type = 0 : character string, maximum length 31 characters, conforming to "C-type" parameter values.

DS type = 1 : one octet containing a parameter type (see 3.3.6) or 2 or more octets containing nested parameter types, in the case of parameters recursively encoded within parameters.

DS type = 2 : one octet containing a message type (see 3.3.5).

Note 4

Some specific service standards may use a special encoding of this parameter in some cases. This use is permitted. Later versions of such standards are expected to remove this special encoding. The version number will avoid ambiguity arising.

3.3.7.2 Encoding of Parameter pp-protocol-definition to PP-CONNECT-REQ (CNQ) and PP-CONNECT-RESP (CNR)

The value of this parameter is an aggregate consisting of fields as defined below:

```
pp-service-ident: single octet, S type value:
               1: "DPS", 2: "VTS".
               0: implies "void", uncommitted.
                  Other values reserved.
```

This field is mandatory; "void", value 0, is used if it is not required to designate a specific service ident in an instance of the parameter.

pp-service/class-version-idents: single octet, M type value:

```
              f1: version 1; f2: version 2; fn: version n;
                  (n < 8 : f8 is reserved for future use
                  as an extender and is set to 0).
              Version is applicable to service-class if
              present, otherwise to service.
```

This field is mandatory; "void", a single octet of all 0's, is used if it is not required to designate a specific service or class version in an instance of the parameter.

pp-service/class-subset-ident: single octet, M type value:

```
         f1 - f4: designate service or class subsets  as
                  defined in specific service standards.
         f5 - f6: specific service dependent.
         f7     : = 1 : field value is "void"; other bits
                  set to 0.
         f8     : 1 : extension subset octet follows,
                  0 : subset field is complete.
         Subset is applicable to service-class if present,
         otherwise to service.
```

This field is mandatory; "void" is used if it is not required to designate a specific subset in an instance of the parameter.

pp-service-class-ident: sincle octet, S type value: This field is optional unless pp-pe-profile is required and if present follows the pp-service/class-subset-ident field; values are as defined in specific service standards. Value 0 of this field is reserved as "void". If this field is present and not "void" the subset, if relevant, applies to the class.

pp-pe-profile: variable length, A type value: This field is optional and if present follows the pp-service-class-ident field which is then mandatory (value may be "void"); values are as defined in specific service or class standards. Value 0 is reserved as "void".

Additional information

In this version the above fields are defined as single octets except where stated. A later version may permit other fields to become variable length using the reserved extension bits. Further information on use is given in sub-clause 3.3.6.1.

3.3.8 Values of Severity in Diagnostic Parameters

Values are defined as follows:

0 : success (default if diagnostic not present)
1 : success with warning
2 : reserved
3 : failure

Other values are available for specific service use.

3.3.9 Values of Reason in Diagnostic Parameters

Values are defined as follows: (the number at the extreme right indicates the severity codes associated with the reason).

0 :	no reason provided (default if diagnostic not present)	0
1 :	non-standard reason	1,2,3
2 :	unset parameter value	3
3 :	illegal parameter value	3
4 :	unsupported parameter value	3
5 :	illegally duplicated parameter	3
6 :	illegal parameter type	3
7 :	unsupported parameter type	3
8 :	illegal message type	3
9 :	unsupported message type	3
10:	*(not yet defined)*	
11:	invalid p-user response parameter	3
12:	protocol sequence violation	3
13:	presentation environment incomplete	3

14:	presentation environment storage full	2,3
15:	no pe save capability: exit failed	3
16:	no pe resume information, pe initialized	2
17:	negotiation-enclosure not supported, not entered	2
18:	negotiation not supported	3
19:	illegally duplicated parameter value	3
20:	conflicting parameter value	3
21:	illegal parameter sequence	3

3.4 Session Service Mapping

The Generic Presentation Protocol relies on the Services offered by the Session Service as described in ECMA-75.

3.4.1 Summary of Usage of Session Services

The following list shows the session service request/ response primitive events which are issued for the various protocol messages defined in clause 3.2.

CNQ	S-CONNECT request
CNR	S-CONNECT response
RLQ	S-RELEASE request
RLR	S-RELEASE response
DNQ	S-DISCONNECT request

3.4.2 Connection Mapping

Each presentation-connection uses one and only one session-connection. The session-connection is used by one and only one presentation connection.

Note 5

A future version of this Standard may permit re-use of the same session-connection for multiple consecutive presentation-connections. Multiplexing of multiple concurrent presentation connections onto a single session-connection is excluded in ISO 7498.

3.4.3 Session Connection Establishment

The establishment of a session-connection between the two presentation-entities is necessary before any presentation protocol activity can take place.

The establishment phase of the presentation-connection is mapped into the establishment facility of Session Layer, the PP-CONNECT-REQ (CNQ) protocol message being mapped (in part) into the SS-user-data parameter of the S-CONNECT request primitive and indication primitive, and the PP-CONNECT-RESP (CNR) protocol message being mapped (in part) into the SS-user-data parameter of the S-CONNECT response and confirmation primitives.

The following parameters of CNQ and CNR are not mapped into SS-user-data but are mapped into other parameters of

S-CONNECT primitives as in ECMA-75:

- pp-user-caller into "initiator address"
- pp-user-called into "acceptor address"
- pp-session-subset into "subset choice"
- pp-session-params into "subset parameters" as relevant.

The result parameter of S-CONNECT response primitive is derived from pp-diagnostic parameter in CNR; if severity value is "failure" result is "rejected", otherwise it is "accepted".

The message header and all remaining parameters of CNQ and CNR are encoded transparently into SS-user-data as defined in 3.3.

3.4.4 Session Connection Termination

The two forms of presentation protocol structure for termination of presentation-connection (arising from the three forms of presentation service element) are mapped into the similar facilities of the session service as follows.

3.4.4.1 Mapping of P-RELEASE

The PP-RELEASE-REQ (RLQ) protocol message is mapped into the SS-user-data parameter of the S-RELEASE request and indication primitives, and the PP-RELEASE-RESP (RLR) protocol message is mapped into the SS-user-data parameter of the S-RELEASE response and confirmation primitives.

The result parameter of S-RELEASE response primitive is derived from p-diagnostic parameter in RLR; if the severity value is "failure" result will be "negative", otherwise it will be "affirmative". The "failure"/"negative" case is valid only if the session subset in use defines the session terminate-token.

The message header and all remaining parameters of RLQ and RLR are encoded transparently into SS-user-data as defined in 3.3.

3.4.4.2 Mapping of P-DISCONNECT

The PP-DISC (DNQ) protocol message is mapped into the SS-user-data parameter of the S-DISCONNECT request and indication primitives. Presentation protocol message type code is not used for DNQ as no other protocol message is mapped into the above session primitives.

The parameter of DNQ is encoded transparently into SS-user-data as defined in 3.3.

3.4.5 Other Session Service Facilities

In this version of this Standard no other session facility is used explicitly. A later version may include possible use of other session facilities to support certain service facilities.

Specific presentation protocols operating within the framework of the Generic Presentation Protocol may themselves define the use of other specialised session facilities and will define explicitly the mapping of the relevant specific presentation protocol messages into these facilities.

All other protocol messages required by a specific protocol will be mapped into the SSDU parameter of the S-DATA facility, or, if provided by the session subset and when applicable to the specific protocol, into the XSSDU parameter of S-EXPEDITED facility.

APPENDICES

APPENDIX A

BRIEF DESCRIPTION OF THE REFERENCE MODEL FOR OPEN SYSTEMS INTERCONNECTION

A.1 Scope

This appendix is a copy of ISO/TC97/SC16 N 575.

This appendix provides a brief description of the Reference Model of Open Systems Interconnection.

A.2 General Description

A.2.1 Introduction

The Reference Model of Open Systems Interconnection provides a common basis for the co-ordination of the development of new standards for the interconnection of systems and also allows existing standards to be placed within a common framework. The model is concerned with systems comprising terminals, computers and associated devices and the means for transferring information between these systems.

A.2.2 Overall Perspective

The model does not imply any particular systems implementation, technology or means of interconnection but rather refers to the mutual recognition and support of the standardized information exchange procedures.

A.2.3 The Open Systems Interconnection Environment

Open Systems Interconnection is not only concerned with the transfer of information between systems (i.e. with communication), but also with the capability of these systems to interwork to achieve a common (distributed) task. The objective of Open Systems Interconnection is to define a set of standards which allow interconnected systems to co-operate.

The Reference Model of Open Systems Interconnection recognizes three basic constituents (see Figure A1):

a) application processes within an OSI environment

b) connections which permit information exchange,

c) the systems themselves.

Note

The application processes may be manual, computer or physical processes.

A.2.4 Management Aspects

Within the Open Systems Interconnection architecture there is a need to recognize the special problems of initiating, terminating, monitoring on-going activities and assisting in their harmonious operations as well as handling abnormal conditions. These have been collectively considered as the management aspects of the Open Systems Interconnection architecture. These concepts are essential to the operation of the interconnected open systems and therefore are included in the comprehensive description of the Reference Model.

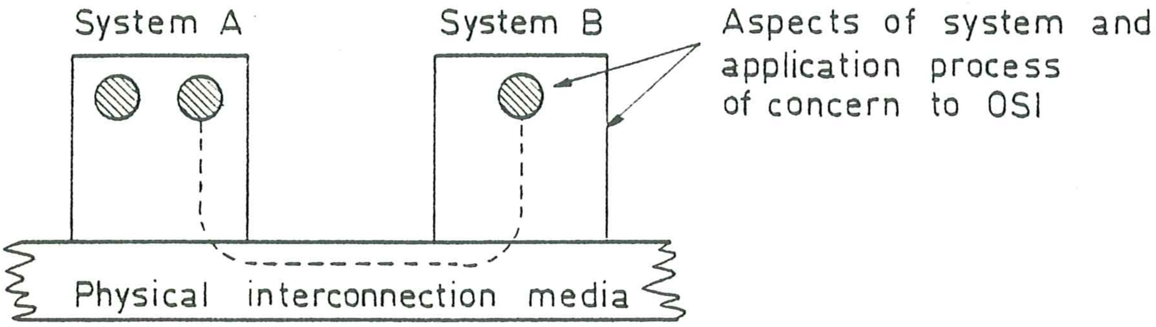

System A
System B
Aspects of system and application process of concern to OSI
Physical interconnection media

Fig. A1 - General Schematic Diagram Illustrating the Basic Elements of Open Systems Interconnection

A.2.5 Concepts of a Layered Architecture

The Open systems architecture is structured in layers. Each system is composed of an ordered set of subsystems represented for convenience by layers in a vertical sequence. Adjacent subsystems communicate through their common interface.

A layer consists of all subsystems with the same rank. The operation of a layer is the sum of the co-operation between entities in that layer. It is governed by a set of protocols specific to that layer.

The services of a layer are provided to the next higher layer, using the functions performed within the layer and the services available from the next lower layer.

An entity in a layer may provide services to one or more entities in the next higher layer and use the services of one or more entities in the next lower layer.

A.3 The layered Model

The seven-layer Reference Model is illustrated in Fig. A2.

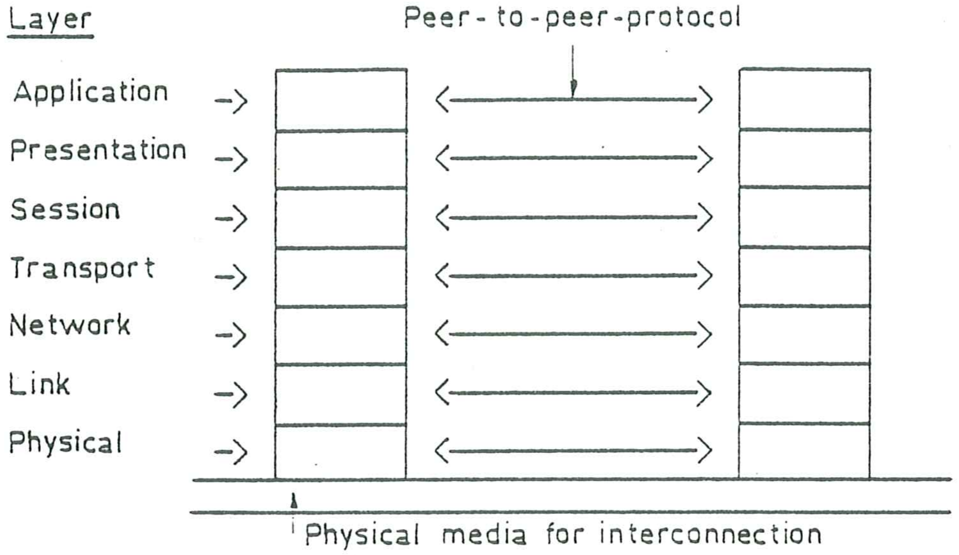


Fig. A2 - The Seven-Layer Reference Model and Peer-to-peer Protocol

A.3.1 The Application Layer

As the highest layer in the Reference Model of Open Systems Interconnection, the Application Layer provides services to the users of the OSI environment, not to a next higher layer.

The purpose of the Application Layer is to serve as the window between communicating users of the OSI environment through which all exchange of meaningful (to the users) information occurs.

The user is represented by the application-entity to its peer.

All user specifiable parameters of each communications instance are made known to the OSI environment (and, thus, to the mechanisms implementing the OSI environment) via the Application Layer.

A.3.2 The Presentation Layer

The purpose of the Presentation Layer is to represent information to communicating application-entities in a way that preserves meaning while resolving syntax differences.

The nature of the boundary between the Application Layer and the Presentation Layer is different from the nature of other Layer boundaries in the architecture.

The following principles are adopted to define a boundary between the Application Layer and the Presentation Layer.

a) Internal attributes of the virtual resource and its manipulation functions exist in the Presentation Layer;

b) external attributes of the virtual resource and its manipulation functions exist in the Application Layer;

c) the functions to use the services of the Session Layer effectively exist in the Presentation Layer;

d) the functions to use services of the Presentation Layer effectively exist in the Application Layer.

A.3.3 The Session Layer

The purpose of the Session Layer is to provide the means necessary for cooperating presentation-entities to organize and synchronize their dialogue and manage their data exchange. To do this, the Session Layer provides services to establish a session-connection between two presentation entities, and to support their orderly data exchange interactions.

To implement the transfer of data between the presentation-entities, the session-connection is mapped onto and uses a transport-connection.

A.3.4 The Transport Layer

The Transport Layer exists to provide the transport service in association with the underlying services provided by the supporting layers.

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

The Transport Layer is required to optimize the use of the available communication resources to provide the performance required by each communicating transport user at minimum cost. This optimization will be achieved within the constraints imposed by considering the global demands of all concurrent transport users and the overall limit of resources available to the Transport Layer. Since the network service provides network connections from any transport entity to any other, all protocols defined in the Transport Layer will have end-to-end significance, where the ends are defined as the correspondent transport-entities.

The transport functions invoked in the Transport Layer to provide requested service quality will depend on the quality of the network service. The quality of the network service will depend on the way the network service is achieved.

A.3.5 The Network Layer

The Network Layer provides the means to establish, maintain and terminate network connections between systems containing communicating application-entities and the functional procedural means to exchange network service data units between two transport entities over network connections.

A.3.6 The Data Link Layer

The purpose of the Data Link Layer is to provide the functional and procedural means to activate, maintain and deactivate one or more data link connections among network entities.

The objective of this layer is to detect and possibly correct errors which may occur in the Physical Layer. In addition, the Data Link Layer conveys to the Network Layer the capability to request assembly of data circuits within the Physical Layer (i.e. the capability of performing control of circuit switching).

A.3.7 The Physical Layer

The Physical Layer provides mechanical, electrical, functional and procedural characteristics to activate, maintain and deactivate physical connections for bit transmission between data link entities possibly through intermediate systems, each relaying bit transmission with the Physical Layer.

APPENDIX B

INDEX AND GLOSSARY OF TERMS

B.1 General

The Terminology used in this Standard consists of:

- Terminology defined in the Reference Model for Open Systems Interconnection, ISO 7498,
- Terminology for concepts of Presentation Layer in general and for Virtual Terminal Service and Protocol, in particular which is defined in this Appendix, see B.2.
- Notation terminology, which is defined in Appendix C.

An Index to terms and acronyms is provided in B.3.

B.2 Glossary of Terms

For the purpose of this Standard the following terms have the meaning indicated. A reference is given to material in the main text of the document.

Conceptual Service Interface, CSI (2.1)

As an aid to describing the services offered by the GPP and the purpose of each of the protocol elements, a means by which the p-users could make their requests known to the service is introduced. Since this is in no way a subject of standardization it is termed a conceptual service interface.

Current PE (2.2.1.1, 2.2.8.1)

The Defined PE which is in use in transfer-enclosure.

Defined PE

A PE which is in a completely agreed state such that it forms one particular usable set of presentation parameters.

Destructive [in the context of effects of services]

See Appendix C.

Directed Graph for Parameters (2.2.5.3)

The method used for representing the relationships between presentation parameters; such relationships constrain the process of negotiation of a PE.

Disestablishment Phase (2.2.1.1, 2.2.3)

The transient phase, consisting of a single service element, whereby a presentation-connection is terminated.

Draft PE (2.2.5.2)

The PE being created or amended within negotiation-enclosure during a multiple-interaction negotiation.

Enclosure (2.2.1.1, 2.2.4)

A phase of the p-connection in which activities of a particular or possibly restricted type are performed. All interaction activities initiated within an enclosure are completed before proceeding to the next phase and thus no activities are allowed to "penetrate" into following enclosures. This does not prevent the information resulting from activity in one enclosure being used in a following enclosure.

Enclosure-token (2.2.4)

An optional token which can be used to mediate the issue of requests for a change of enclosure.

Establishment Phase (2.2.1.1, 2.2.2)

The transient phase, consisting of a single service element, whereby a presentation-connection is set up between a pair of peer p-users.

Initial PE (2.2.2, 2.2.5.2)

The PE brought into existence by establishment of a p-connection. In general it will not necessarily be complete, i.e. a Defined PE, but in some GPP subsets it is constrained to be.

Multiple-interaction Negotiation (2.2.1.2, 2.2.7)

The type of negotiation used within a negotiation-enclosure to create or modify a Defined PE. Only one PE can be negotiated by one occurrence of negotiation-enclosure. During the operation presentation parameters can be negotiated individually or in groups. A PE under negotiation does not become a new Defined PE unless and until the negotiation-enclosure is successfully terminated.

Multiple PE Option (2.2.1.1, 2.2.2.2)

The optional capability for simultaneously holding more than one Defined PE definition during a p-connection, although at most one can actually be in use (be the Current PE) at any instant.

Negotiate-token (2.2.5.3)

A token used to mediate which p-user can issue service requests during multiple-interaction negotiation.

Negotiation (2.2.5)

The action of redefining an existing PE or creating a new PE. The ability to create new PEs may be restricted by establishment time options (multiple PE option).

Negotiation-enclosure (2.2.1.1, 2.2.7)

The phase in a p-connection where the p-users may negotiate using multiple-interaction negotiation. The ability to use this type of enclosure is subject to agreement between the two p-users.

Negotiation Group (2.2.5.3)

A set of related parameters currently being negotiated.

Null-enclosure (2.2.1.1)

The type of enclosure entered after establishment of a p-connection if there is not a Defined PE, or at other times when the users so desire; there is no Current PE in null-enclosure. The ability to use this type of enclosure is subject to agreement between the two p-users.

Parameter [in the context of negotiation] (2.2.5.1)

The objects of a negotiation operation whose values subsequently determine the precise nature of the p-service available on the p-connection. Parameters can be grouped hierarchically to form sets of related parameters using the directed graph concept.

Presentation Environment, PE (2.2.1, 2.2.5.1)

A complete and self-consistent set of values for the parameters of the p-service. It includes any parameter specific to a particular p-service operating within the framework of the GPP. It is not necessary for all possible presentation parameters to have values to complete a particular PE. Depending on establishment time options one or more PEs can be in defined state (Defined PE) at the same time on one p-connection. At most one PE can be in use (Current PE). All PEs cease to exist when the p-connection is disestablished.

Protocol Message (3.1.1)

The smallest unit of information transferred across the Open Systems Interconnection having semantic significance to the GPP p-entity.

Root Parameter (2.2.5.3)

The highest order parameter of a negotiation group, i.e. nearest the head of the directed graph.

Sequenced [in the context of effects of services]

See Appendix C.

Service Element (2.1)

The smallest unit of activity of the Service with self contained semantic significance which can be initiated by one service-user and which may or may not have a consequential effect on another service-user. A service element consists of one or more service primitives issued by or received by one or both service users.

See also Appendix C.

Service Primitive (2.1)

The smallest unit of activity on the conceptual service interface at one service access point; one or more service primitives at one or both SAPs make up a service element. Note that there is neither a one-to-one nor indeed any specific mapping implied between these service primitives and protocol messages. Only primitives of the conceptual service interface are referred to in this Standard. Rules for defining real service primitives and for use of macros as part of a real service interface are not the concern of this Standard.

See also Appendix C.

Single-interaction Negotiation (2.2.1.1, 2.2.6)

The negotiation consisting of only a single confirmed service element, available in either null-enclosure or transfer-enclosure; no change of enclosure can be effected by it.

Token (2.1.8, 2.2.8.1)

An abstract object with which is associated some capability available to a pair of peer entities, and which mediates the actual use of the capability to avoid collision situations. Any token is owned at any instant by at most one of the entities. A token of the presentation layer may be a service-token visible to the p-users and mediating their activities, or a protocol-token used by the p-entities to meditate protocol activities and not directly visible to the p-users.

Transfer-enclosure (2.2.1.1, 2.2.8.1)

The type of enclosure in which the p-users may operate on data and control information (as distinct from parameters, see negotiation-enclosure). This will be performed within the context of a selected Defined PE, the Current PE.

B.3 Index of Acronyms

AC	Access Control
BNF	Backus-Naur Format
CCA	Conceptual Communications Area
CDS	Conceptual Data Store
CICI	Conceptual Interface Command Interpreter
CSS	Control, Signalling and Status (applied to area or mechanism)
DSD	Data Structure Definition
GPP	Generic Presentation Protocol
GPS	Generic Presentation Service
GVT	Generic Virtual Terminal
II	Indication-Indication (applied to service structure)

PE	Presentation Environment
PPDU	Presentation Protocol Data Unit
PSAP	Presentation Service Access Point
RC	Request-Confirmation (applied to service structure)
RI	Request-Indication (applied to service structure)
RO	Request Only (applied to service structure)
SSDU	Session Service Data Unit
TLV	Type, Length, Value (applied to encoding technique)
TWA	Two-Way Alternate (as a dialogue discipline)
TWS	Two-Way Simultaneous (as a dialogue discipline)
VTP	Virtual Terminal Protocol
VTS	Virtual Terminal Service

APPENDIX C

SERVICE DESCRIPTION TECHNIQUE

The Service Description is in terms of a number of "Service Elements". A Service Element is an elementary operation of the Service viewed as a whole, i.e. taking account of both the Service Access Points at which the service-users interact with the service. A Service Element thus consists of one or more events on the conceptual service interface at one or both service access points. These events are known as service primitives, frequently abbreviated to just primitive. Primitives, depending on the defined structure of the service element, can occur in one or more of the event forms "request" "indication" "response" "confirmation". Parameters will usually be associated with the primitives, and may be supplied by the issuing layer-entity or "filled-in" by the receiving layer entity for the use of the issuer.

The four types of service elements (known as service structures) are given mnemonic names in this Standard as follows:

- RI Type 1, i.e. Request-Indication
- RC Type 2, i.e. Request-Indication-Response-Confirmation
- II Type 3, i.e. Indication-Indication
- RO Type 4, i.e. Request-Only.

For each of the service elements relevant to a particular part of the total presentation-service the following information is given in the sub-clauses of the Service Description:

- the name of the primitive (as title),
- the purpose of the primitive,
- the type(s) of service elements (and hence the applicable event forms in which the service element can occur),
- a table showing the applicable forms of the primitives and the parameters which can accompany the primitive for each applicable form, with the entries for each listed parameter conforming to the following key:
  - D : the parameter is supplied by the user,
  - U : the parameter is supplied by the service,

  - D,U: the U form is the combination of the user-supplied parameter (D) and the service-supplied parameter,
  - x : the parameter is not used in the particular form of the primitive,
  - (n) : indicates that the parameter can be multi-valued,
  - (1) : indicates that the parameter must be single-valued (usually shown in response to a multi-valued request parameter). (1) is the default if no entry is made.
- notes on the use of the primitive and/or parameters,
- effects of the primitive and effects, if any, on other primitives, issued before and after the event(s) of this primitive,
- additional information as necessary.

Where effects are described as sequenced this means, as a general rule, that with respect to other services described as sequenced, indications appear in the same sequence as requests, and confirmations appear in the same sequence as responses (where the service structures contain these elements). The protocol definition will determine whether or not response/confirmation pairs are always in the same sequence as their associated request/ indication pairs - this is not necessarily implied by the term sequenced. Any relative sequencing not covered by this general rule will be specified explicitly with the service element description.

Where effects are described as destructive this means that events (including data delivery) associated with previously initiated services may not occur.

APPENDIX D

ADDITIONS TO STANDARD ECMA-75 SESSION PROTOCOL

The following material is an extract from a document which gives an illustrative specification of the additional facilities to be incorporated in a future version of ECMA-75 to provide the session service subset on which this Basic Class Virtual Terminal Service is mapped. These additions are not necessarily the only ones which will be considered for the next version of ECMA-75 and the final definitive specification of these additions in the future version of ECMA-75 is not to be inferred from the following illustrative description.

D.1 Introduction

The consolidation of the work in the definition of Basic Class Virtual Terminal Service and Protocol has led to clearly determine its requirements on the session service.

According to these requirements a new service primitive has been defined (called S-TYPED-DATA) and a new service subset ("E": Basic Terminal Subset) has been introduced.

The parameter of the S-DISCONNECT primitive has been extended in order to take advantage of the increased user field available in the transport expedited service (2nd edition of ECMA-72).

This Appendix lists the changes in the service, protocol and formal description of ECMA-75, that are needed to implement the above decisions. In addition, the S-TYPED-DATA primitive has been added to subset C.

Note D.1

The definition of the new subset E that includes the newly defined S-TYPED-DATA primitives enables VTP to make full use of ECMA-75 at least for its initial requirements. At this stage of analysis, the adoption of subset E as the underlying session service for basic class VTP does not seem to preclude compatible enhancements (especially towards subset C) for the more sophisticated terminal protocols that will be defined in the future.

D.2 Changes of the service part of ECMA-75

a) p.8, 5th line, after confirmation add: "This restriction does not hold for typed data" (see 1.5.13).

b) p.13, add:

"1.5.13 Typed data

During the lifetime of a session connection it might be necessary for the session users to exchange information that, by their very nature should not be constrained by any token management function.

In what follows, these data will be indicated as typed data".

<u>*Note D.2*</u>

A typical use of this feature is to transfer information regarding the SS-user status and/or the control of the session connection itself.

c) p.17 in the table, right below S-DATA, add:

S-TYPED DATA	request indication

d) p.19, after 2.2.1.5, add

"2.2.1.6 <u>S-TYPED-DATA primitives</u>

The S-TYPED-DATA primitives are a means for an SS-user to transfer a typed SSDU to the other user.
The effects are sequentially transmitted and non-disruptive.
The service structure is type 1. Parameters are defined in 2.2.2.5. The right to be initiator is not subject to token assignments .

<u>*Note D.3*</u>

In particular, where S-DATA and SSDU's are transmitted in the same direction the sequence of SSDU's is preserved".

<u>*Note D.4*</u>

A limit will be defined for the length of the S-TYPED-DATA primitive. This limit is the subject of further study.

e) p.19 renumber sections 2.2.1.6 to 2.2.1.13 accordingly.

f) p.22, in Rule 10 add:

"This maximum size holds both for SSDUs transmitted as S-DATA and SSDUs transmitted as S-TYPED-DATA".

g) p.23, replace the Rule 1 with:

"<u>Rule 1</u>. The ss-user data is transparent to the session service. The maximum size is eleven octets".

h) p.24, replace in all section 2.2.2.5 "S-DATA" with "S-DATA and S-TYPED-DATA".

i) p.28, third line: change "Four subsets" with "the following subsets".

j) p.28, after the subset list, add:

"Subset E - basic terminal subset".

k) p.30, sect. 2.3.4.2 in the list after S-DATA, add:

"- S-TYPED-DATA primitives"

l) p.31 Rule 9 at the end add:

"In any case, SSDUs which arise from S-DATA and from S-TYPED-DATA primitives will not be blocked together".

m) p.32, add at the end:

"2.3.6 Subset E: Basic Terminal Subset

2.3.6.1 Purpose

Subset E provides to ss-users a synchronized dialogue structure and a TWA interaction with the ability to exchange information independently of the data token.

2.3.6.2 Content

The service primitives included in this subset are:

- S-CONNECT primitives
- S-RELEASE primitives
- S-DISCONNECT primitives
- S-ABORT primitives
- S-DATA primitives
- S-TYPED-DATA primitives
- S-SYNC primitives
- S-TOKENS-GIVE primitives
- S-PLEASE primitives

The selection of all these primitives and their associated services is implicit in the choice of this subset (i.e. value "E" in the subset choice parameter of S-CONNECT, see 2.2.2.1). Interaction type is TWA (i.e. the Data-token is always defined and assigned). Minor-synchronization-points are supported (i.e. the synchronize-token is always defined and assigned). Negotiated termination is supported (i.e. the Terminate-token is always defined and assigned).

The additional parameters in S-CONNECT which are specific to this subset and their values are defined in Table 2.3/4 and the following rules which reference it.

Table 2.3/4 - S-CONNECT, Parameters specific to subset E

Parameter Name	Rep.	Ind.	Resp.	Conf.
Data token	D1	U	D2	U
Synchronize token	D1	U	D2	U
Terminate token	D1	U	D2	U
Initial serial number	D3	U	D4	U

Rule 1 : value is "initiator" or "acceptor" or "acceptor chooses".

Rule 2 : if the value in the request/indication is "acceptor chooses" this value is either "initiator" or "acceptor". Otherwise the assignment in the request/indication is not negotiable and this value is the same.

Rule 3 : this is any value in the range defined in 1.5.5.

Rule 4 : this value takes precedence if it is different from that in the request/indication. It is any value in the range defined in 1.5.5."

D.3 Changes of the protocol part of ECMA-75

a) p. 43, section 3.2.3.4, in the paragraph headed "Content":

In the third line, replace "(3 octets)" with "(11 octets)".

b) p. 43, section 3.2.4, second sentence.

This sentence should read:

"It uses the DATA TRANSFER (DT), TYPED DATA (TD), MARK CONFIRMATION (MC) and EXPEDITED (EX) messages."

c) p. 44, section 3.2.4.1, first 8 lines.

These lines should read:

"3.2.4.1 The DATA TRANSFER and TYPED DATA messages

Function

The DATA TRANSFER (DT) and TYPED DATA (TD) messages are used to carry session user data and delimiting information between session connection users. They are also used to give tokens. If the data token has been defined for that session connection, only the owner of this token may issue DT. TD can be issued independent of the tokens. If the synchronize token ..."

d) p. 45, section 3.2.4.1.

Add new sentence after first sentence in paragraph headed "Sending":

"A valid "SPM service TYPED TRANSFER request" event (EVE 4D), results in a TYPED DATA message."

e) p. 45, section 3.2.4.1.

Add new sentence after first sentence in paragraph headed "Receiving":

"A valid "incoming TYPED TRANSFER message" event (EVE 15D) normally results in an "SPM service TYPED TRANSFER indication" event (EVE 104D)."

f) p. 44, section 3.2.4.1, Mark Type C.

This paragraph should read:

"Type C This mark supports the service of minor synchronization points (urgent type). A MARK CONFIRMATION (MC) is required before all subsequent transmission on the transport normal flow, except that TYPED DATA (TD) may be issued before receipt of MC. There is no other implication on the data flows."

g) p. 51, section 3.2.5.4, last sentence.

For : "i.e. a DATA TRANSFER message,"
Read: "i.e. a DATA TRANSFER or TYPED DATA message,"

h) p. 58, table 3.4/3 - Encoding Fixed Header.

Add a new SPDU category:

VALUE	MEANING
6	TYPED DATA category

i) p. 59, table 3.4/4 - messages/SPDU category correspondance.

Add a new entry in the table:

Message Name	Category
TYPED DATA	TYPED DATA

i) p. 61, table 3.4/6 - Items defined in each SPDU category.

Modify the entry DATA so that it becomes:

SPDU category	Item	TLV Format	Type (see Note 40)
DATA or TYPED DATA	Session - TLVed User data	F3 (see Note 41)	16 (see Note 41)
	Mark/Token extension	F3	17

k) p. 62, section 3.4.6.1.

Modify the title: "3.4.6.1 DATA TRANSFER and TYPED DATA messages"

Table 3.4/9: in the SPDU category replace "DATA" with "DATA or TYPED DATA" (five times).

1) p.71, section 3.4.6.13.

In the paragraph headed "Information Unit", third line, replace "(maximum size is 3 octets)" with "(maximum size is 11 octets)".

m) p. 74, section 3.5.4.

After "-MARK CONFIRMATION", insert new line:

"-TYPED DATA"

n) p. 74, section 3.5.5.

Add after 3.5.5 a new section:

"3.5.6 Subset E: Basic Terminal Subset

The SPDU protocol messages included in this subset are:

- CONNECT
- ACCEPT
- REFUSE
- FINISH
- DISCONNECT
- ABORT
- DATA TRANSFER
- TYPED DATA
- PLEASE TOKENS
- GIVE TOKENS
- MARK CONFIRMATION
- NOT FINISHED

Parameters and parameter values which relate to services and messages that are not included in the subset are not used.

The TSM protocol is defined in section 3.3."

D.4 Changes of the formal description of ECMA-75 (p. 97 to p. 111)

Extensions are needed in order to integrate the TYPED DATA message in the formal description of ECMA-75.

The extensions are mainly:

- new events (SPM-driving events and SPM generated events)
- new transitions (between these new events and existing states).

They will be provided in the revision of ECMA-75.