INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

X.881

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(07/94)

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

OSI APPLICATIONS – REMOTE OPERATIONS

INFORMATION TECHNOLOGY – REMOTE OPERATIONS: OSI REALIZATIONS – REMOTE OPERATIONS SERVICE ELEMENT (ROSE) SERVICE DEFINITION

ITU-T Recommendation X.881

(Previously “CCITT Recommendation”)

Foreword

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. Some 179 member countries, 84 telecom operating entities, 145 scientific and industrial organizations and 38 international organizations participate in ITU-T which is the body which sets world telecommunications standards (Recommendations).

The approval of Recommendations by the Members of ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, 1993). In addition, the World Telecommunication Standardization Conference (WTSC), which meets every four years, approves Recommendations submitted to it and establishes the study programme for the following period.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC. The text of ITU-T Recommendation X.881 was approved on 1st of July 1994. The identical text is also published as ISO/IEC International Standard 13712-2.

NOTE

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

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS
(February 1994)

ORGANIZATION OF X-SERIES RECOMMENDATIONS

Subject area	Recommendation series
PUBLIC DATA NETWORKS	
Services and facilities	X.1-X.19
Interfaces	X.20-X.49
Transmission, signalling and switching	X.50-X.89
Network aspects	X.90-X.149
Maintenance	X.150-X.179
Administrative arrangements	X.180-X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200-X.209
Service definitions	X.210-X.219
Connection-mode protocol specifications	X.220-X.229
Connectionless-mode protocol specifications	X.230-X.239
PICS proformas	X.240-X.259
Protocol identification	X.260-X.269
Security protocols	X.270-X.279
Layer managed objects	X.280-X.289
Conformance testing	X.290-X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300-X.349
Mobile data transmission systems	X.350-X.369
Management	X.370-X.399
MESSAGE HANDLING SYSTEMS	X.400-X.499
DIRECTORY	X.500-X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600-X.649
Naming, addressing and registration	X.650-X.679
Abstract Syntax Notation One (ASN.1)	X.680-X.699
OSI MANAGEMENT	X.700-X.799
SECURITY	X.800-X.849
OSI APPLICATIONS	
Commitment, concurrency and recovery	X.850-X.859
Transaction processing	X.860-X.879
Remote operations	X.880-X.899
OPEN DISTRIBUTED PROCESSING	X.900-X.999

Contents

Page

Summary

This Recommendation | International Standard provides the OSI realizations of the abstract concepts of the Remote Operations Service (ROS). It describes the services provided by ROSE and the mappings of these services onto those provided by the Association Control Service Element (ACSE), the Reliable Transfer Service Element (RTSE) and the presentation layer.

Introduction

Remote operations (ROS) is a paradigm for interactive communication between objects. As such it can be used in the design and specification of distributed applications. The basic interaction involved is the invocation of an operation by one object (the invoker), its performance by another (the performer), possibly followed by a report of the outcome of the operation being returned to the invoker.

The concepts of ROS, as specified in ITU-T Rec. X.880 | ISO/IEC 13712-1, are abstract and may be realized in many ways. For example, objects whose interactions employ ROS concepts may be separated by a software interface or by an OSI network.

This Recommendation | International Standard provides the framework for the realization of an operation package and association contract as an OSI application context. Such an application context is specified primarily in terms of a collection of application service elements (ASE). From a ROS perspective, these ASEs fall into three broad categories:

a) operation-specific ASEs, which embody knowledge of the definitions of the operations in the association contract;

b) the Remote Operations ASE (ROSE) which drives the general-purpose protocol required to invoke and report returns of arbitrary operations;

c) information transfer ASEs concerned with the establishment and release of associations where necessary, and the communication of the ROSE protocol control information (PCI). In the OSI realization, such ASEs are the Association Control Service Element (ACSE), the Reliable Transfer Service Element (RTSE) used together with the services of the Presentation layer.

This Recommendation | International Standard focuses on the derivation of ROSE-based application context specifications, the service provided by ROSE, and the way that ROSE is used. This Recommendation | International Standard is a revision of CCITT Rec. X.219 | ISO/IEC 9072-1. The existing usage of ROSE in connection with ACSE, RTSE and the Presentation layer as defined in CCITT Rec. X.219 | ISO/IEC 9072-1 remains valid after this revision. This revision makes no change to the ROSE PCI.

Annex A forms an integral part of this Recommendation | International Standard.

Annexes B and C do not form an integral part of this Recommendation | International Standard.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – REMOTE OPERATIONS: OSI REALIZATIONS – REMOTE OPERATIONS SERVICE ELEMENT (ROSE) SERVICE DEFINITION

1 Scope

This Recommendation | International Standard provides the framework for the realization as an OSI application context of the abstracts concepts of operation package and association contract defined in ITU-T Rec. X.880 | ISO/IEC 13712-1. Such an application context is described in terms of a collection of application service elements, in particular the Remote Operations Application Service Element (ROSE), which drives the general purpose protocol for invoking and reporting the returns of arbitrary operations.

The terms, definitions, and mechanisms defined in ITU-T Rec. X.880 | ISO/IEC 13712-1 apply here and are specialized for an OSI realization as specified in this Recommendation | International Standard. This Recommendation | International Standard focuses on the services provided by ROSE, and the way ROSE is used. The ROSE services are provided by the use of the ROSE protocol (specified in a companion Recommendation | International Standard, ITU-T Rec. X.882 | ISO/IEC 13712-3), in conjunction with the Association Control Service Element (ACSE) services (ITU-T Rec. X.217 | ISO/IEC 8649) and the ACSE protocol (ITU-T Rec. X.227 | ISO/IEC 8650-1), and, optionally, the Reliable Transfer Service Element (RTSE) services (ITU-T Rec. X.218 | ISO/IEC 9066-1) and the RTSE protocol (ITU-T Rec. X. 228 | ISO/IEC 9066-2), and the Presentation service (ITU-T Rec. X.216 | ISO/IEC 8822).

No requirement is made for conformance to this Recommendation | International Standard.

2 Normative references

The following ITU-T Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Specification. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Specification are encouraged to investigate the possibility of applying the most recent editions of the Recommendations and Standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunications Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.
- ITU-T Recommendation X.681 (1994) | ISO/IEC 8824-2:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.
- ITU-T Recommendation X.682 (1994) | ISO/IEC 8824-3:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification*.
- ITU-T Recommendation X.683 (1994) | ISO/IEC 8824-4:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*
- ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The basic model*.
- ITU-T Recommendation X.210 (1993) | ISO/IEC 10731:1994, *Information technology – Open Interconnection – Conventions for the definition of OSI services*.
- ITU-T Recommendation X.215 (1994) | ISO/IEC 8326:1994, *Information technology – Open Systems Interconnection – Session service definition*.
- ITU-T Recommendation X.216 (1994) | ISO/IEC 8822:1994, *Information technology – Open Systems Interconnection – Presentation service definition*.

- ITU-T Recommendation X.217 (1995) | ISO/IEC 8649:1995, *Information technology – Open Systems Interconnection – Service definition for the Association Control Service Element*.
- ITU-T Recommendation X.227 (1995) | ISO/IEC 8650-1:1995, *Information technology – Open Systems Interconnection – Connection-oriented protocol for the Association Control Service Element: Protocol specification*.
- ITU-T Recommendation X.880 (1994) | ISO/IEC 13712-1:1995, *Information technology – Remote Operations: Concepts, model and notation*.
- ITU-T Recommendation X.882 (1994) | ISO/IEC 13712-3:1995, *Information technology – Remote Operations: OSI realizations – Remote Operations Service Element (ROSE) protocol specification*.

2.2 Paired Recommendations | International Standards equivalent in technical content

- ITU-T Recommendation X.218 (1993), *Reliable Transfer: Model and service definition*.

  ISO/IEC 9066-1:1989, *Information processing systems – Text communication – Reliable Transfer – Part 1: Model and service definition*.
- ITU-T Recommendation X.228 (1993), *Reliable Transfer – Protocol specification*.

  ISO/IEC 9066-2:1989, *Information Processing Systems – Text communication – Reliable Transfer – Part 2: Protocol specification*.
- CCITT Recommendation X.219 (1988), *Remote Operations: Model, notation and service definition*.

  ISO/IEC 9072-1:1989, *Information processing systems – Text communication – Remote Operations – Part 1: Model, notation and service definition*.
- CCITT Recommendation X.229 (1988), *Remote Operations: Protocol specification*.

  ISO/IEC 9072-2:1989, *Information processing systems – Text communication – Remote Operations – Part 2: Protocol specification*.

3 Definitions

3.1 Reference Model definitions

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.200 | ISO/IEC 7498-1:

a) abstract syntax;

b) Application Layer;

c) application-process;

d) application-entity;

e) application-service-element;

f) application-protocol-data-unit;

g) application-protocol-control-information;

h) Presentation Layer;

i) presentation-service.

3.2 Service conventions definitions

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.210 | ISO/IEC 10731:

a) service-provider;

b) service-user;

c) confirmed service;

d) non-confirmed service;

e) provider-initiated service;

f) service-primitive; primitive;

g) request (primitive);

h) indication (primitive);

i) response (primitive); and

j) confirm (primitive).

3.3 Presentation service definitions

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.216 | ISO/IEC 8822:

a) abstract syntax name;

b) transfer syntax name;

c) presentation context.

3.4 Association control definitions

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.217 | ISO/IEC 8649:

a) application-association; association;

b) application context;

c) Association Control Service Element.

3.5 Reliable Transfer definitions

This Recommendation | International Standard makes use of the following terms defined in CCITT Recommendation X.218 | ISO/IEC 9066:

– Reliable Transfer Service Element.

3.6 ROSE definitions

For the purpose of this Recommendation | International Standard the following definitions apply:

3.6.1 association-initiating-application-entity; association-initiator: The application-entity that initiates the application-association.

3.6.2 association-responding-application-entity; association-responder: The application-entity that responds to the initiation of an application-association by another AE.

3.6.3 invoking-application-entity; **invoker**: The application-entity that invokes the Remote Operation.

3.6.4 performing-application-entity; performer: The application-entity that performs a Remote Operation invoked by the other application-entity.

3.6.5 requestor: The part of an application-entity that issues a request primitive for a particular ROSE service.

3.6.6 acceptor: The part of an application-entity that receives the indication primitive for a particular ROSE service.

3.6.7 linked operation: See 3.3.8 of ITU-T Rec. X.880 | ISO/IEC 13712-1.

3.6.8 parent-operation: An operation during whose execution the performer may invoke linked-operations.

3.6.9 child-operation: An operation that is invoked by the performer of a parent-operation in response to the latter operation's invocation.

3.6.10 ACSE-user: That portion of an application entity which performs the mapping of the bind operation and unbind operation onto ACSE.

3.6.11 Remote Operation Service Element: The application-service-element defined in this Recommendation | International Standard.

3.6.12 ROSE-provider: The provider of the Remote Operations Service Element services.

3.6.13 ROSE-user: That portion of an application entity which interacts with ROSE for the purpose of communicating with the remotely-located peer user.

3.6.14 RTSE-user: That portion of an application entity which performs the mapping of the bind operation and unbind operation onto RTSE.

4 Abbreviations

AE	Application entity
ACSE	Association Control Service Element
ASE	Application service element
ASN.1	Abstract Syntax Notation One
APDU	Application protocol data unit
RO (or ROS)	Remote Operations
ROSE	Remote Operations Service Element
RT (or RTS)	Reliable Transfer
RTSE	Reliable Transfer Service Element

5 Conventions

This Recommendation | International Standard defines services for the ROSE following the descriptive conventions defined in ITU-T Rec. X.210 | ISO/IEC 10731. In clause 8, the definition of each ROSE service includes a table that lists the parameters of its primitives. For a given primitive, the presence of each parameter is described by one of the following values:

Blank	Not applicable
M	Mandatory
U	User option
C	Conditional
O	Presence is a ROSE service-provider option

In addition, the notation (=) indicates that a parameter value is semantically equal to the value to its left in the table.

This specification employs ASN.1, as specified in ITU-T Rec. X.681 | ISO/IEC 8824-2, to define the **APPLICATION-CONTEXT** information object class. This also provides the notation with which designers of ROS applications can specify particular instances of the class.

6 OSI Realization model for ROS

A general model for the realization of ROS by communication means is shown in Figure 1 (reproduced from Figure 3 of ITU-T Rec. X.880 | ISO/IEC 13712-1).

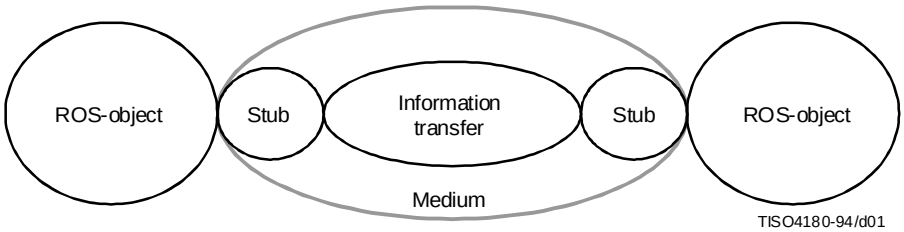
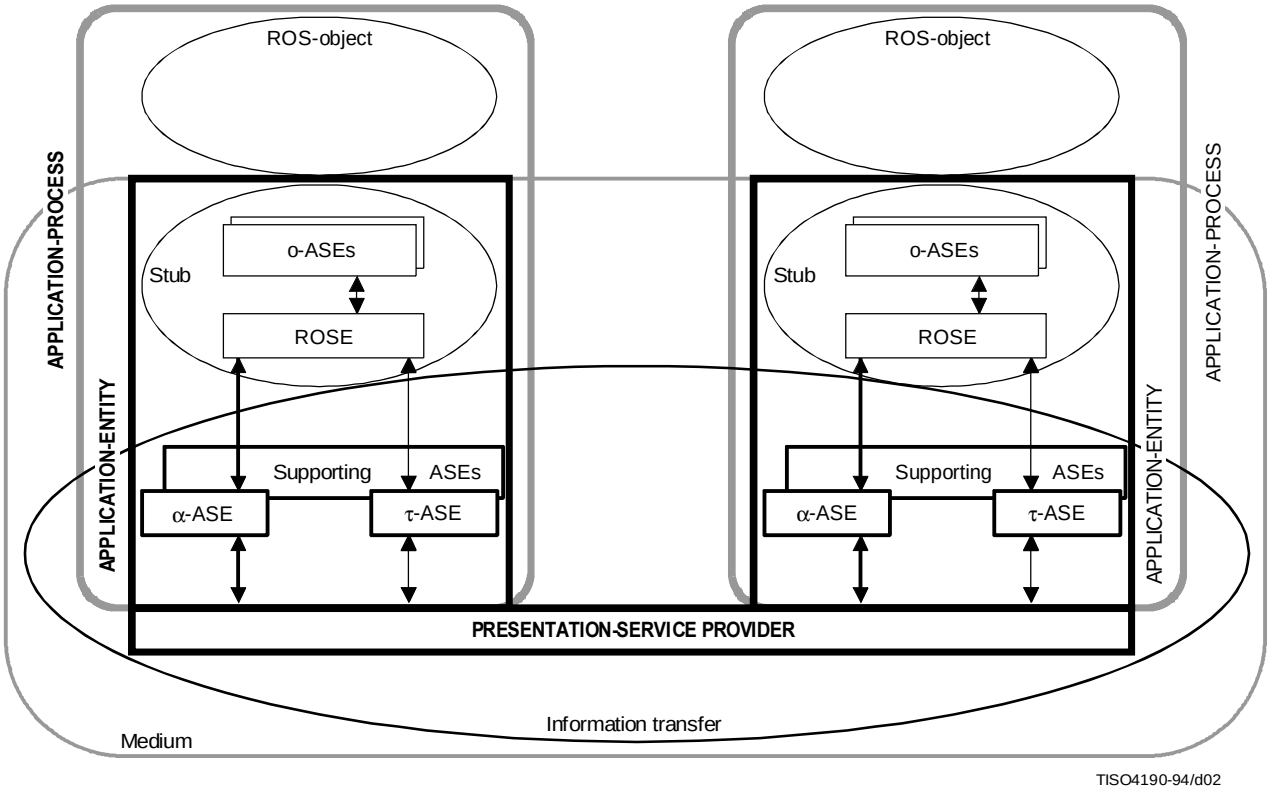


Figure 1 – Communications realization of ROS

Here the stubs represent the ability for the ROS-objects to invoke operations remotely. A particular stub corresponds to the operations in some association contract. The information transfer object conveys protocol data units (PDUs) between the stubs.

This document is concerned with the ROS-objects being realized as application-processes, and the medium by the communications services of OSI.

Figure 2 is a rearranged and expanded version of Figure 1, overlaying on it some of the principal concepts of the application layer of OSI.



α-ASE ASE providing (dynamic) association establishment and release
τ-ASE ASE providing information transfer
ROSE Remote Operations ASE
o-ASEs operation-specific ASEs

Figure 2 – OSI realization of ROS

The stub objects are realized by ROSE together with a collection of operation-specific ASEs. ROSE, whose services are defined in clause 8, drives the generic protocol required to invoke and report returns of arbitrary operations. Each operation-specific ASE embodies knowledge of the definitions of the specific operations involved in some operation package. Where the operation package is asymmetrical, the corresponding operation-specific ASE is also asymmetrical, having a consumer or supplier role to match that of the ROS-object which it is representing. Collectively, ROSE and the operation-specific ASEs have knowledge of all of the operations of the association contract.

The information transfer object is realized by the OSI presentation-service provider, together with a collection of ASEs which shall include an α-ASE, may include a τ-ASE, and may also include ASEs supporting these (e.g. an ASE providing a Directory User Agent function). The collection always include ACSE. Different OSI realizations of ROS result from the use of different collections of these ASEs.

The use of the ROSE services is only possible after a transfer service has been made available on the application association. This transfer service can be available either directly at the level of the Presentation service, or as a service provided by an ASE (see the τ-ASE in Figure 2).

7 ROS-based application contexts

7.1 General

The particular set of ASEs involved in realizing some particular association contract, together with any rules for their coordinated working, constitutes an application context. The application context include all ASEs contributing to the stubs and to the information transfer object.

In realizing the stubs, all application contexts relevant to this Recommendation | International Standard include ROSE. In addition, each such application context includes one operation-specific ASE concerned with each operation package (including one for the connection package, if there is one).

Different application contexts can be defined to realize the same association contract by the use of different sets of ASEs to support information transfer. The information transfer ASEs shall be selected to meet the various Quality of Service requirements inherent in the association contract.

NOTE 1 – It is possible that in the future, rules could be defined for determining which ASEs should be involved in order to meet particular Quality of Service (QOS) requirements. However, it is presently assumed that this is a manual process; that is, the designer of a realization takes these requirements into account and chooses the ASEs accordingly.

All application contexts include ACSE, although this may be acting as the α-ASE or as a supporting ASE.

NOTE 2 – The application context may include additional ASEs for purposes outside of the scope or concern of ROSE, provided that they are arranged to be harmonious with the ASEs mentioned here.

7.2 Application context specification

7.2.1 The static aspects of a ROS-based application context definition can be described as an information object of the class **APPLICATION-CONTEXT**, which is specified as follows:

```
APPLICATION-CONTEXT ::= CLASS
{
        &associationContract                CONTRACT,
        &associationRealization             REALIZATION OPTIONAL,
        &transferRealization                REALIZATION,
        &AbstractSyntaxes                   ABSTRACT-SYNTAX,
```

This definition specifies the ROS aspects of an application context definition. If ASEs other than ROS-based ASEs are used as a part of some application context, the definition in this clause is an element of a composite application context definition.

The manner of defining such a composite application context is outside the scope of this Recommendation | International Standard.

7.2.2 The **&associationContract** field identifies the association contract which this application context realizes.

NOTE – The application designer's intentions regarding whether the “responder can unbind” and “unbind can fail” is derived from the **&associationContract** field.

7.2.3 The **&associationRealization** field shall be present if and only if the **&connection** field of **&associationContract** is present. If present, it shall identify a particular approach to dynamic association establishment and release. A number of such approaches are specified in ITU-T Rec. X.882 | ISO/IEC 13712-3.

7.2.4 The **&transferRealization** field shall identify a particular realization of the information transfer object. A number of such realizations are specified in ITU-T Rec. X.882 | ISO/IEC 13712-3. The **&AbstractSyntaxes** field contains the abstract syntaxes which are required for the conveyance of information between the objects, including the PDUs for invoking and reporting on the operations in the contract. Requirements for these abstract syntaxes are specified in ITU-T Rec. X.882 | ISO/IEC 13712-3. The **&applicationContextName** value shall be used, when the OSI association is being established, to identify the application context which must be in place on this association.

7.3 Relationship with other ASEs and Lower Layer Services

7.3.1 Other Application Service Elements

The ROSE is intended to be used with other ASEs in order to support specific interactive information processing tasks. Therefore, it is expected that the ROSE will be included in a large number of application-context specifications.

The collection of the ROSE and other ASEs included in an application context are required to use the facilities of the presentation-service in a co-ordinated manner among themselves.

The ROSE requires an existing application association controlled by ACSE.

For some application context specifications, a Reliable Transfer Service Element (RTSE) is included.

7.3.2 Presentation Service

If an application context including RTSE and ROSE is defined, ROSE services do not use the Presentation service.

If an application context including ROSE but excluding RTSE is defined, the ROSE services require access to the P-DATA service and require the use of the duplex functional unit of the Presentation service. The ROSE services neither use, nor constrain the use of, any other Presentation service.

Identification of the named abstract syntax in use is assumed for all ROSE services, however this is a local matter and outside the scope of this Recommendation | International Standard.

8 Basic ROSE services

The ROSE services are listed in Table 1.

Table 1 – ROSE services

Service	Type
RO-INVOKE	Non-confirmed
RO-RESULT	Non-confirmed

8.1 RO-INVOKE service

The RO-INVOKE service is used by one ROSE-user (the invoker) to cause the invocation of an operation to be performed by the other ROSE-user (the performer). This service is a non-confirmed service.

The related service structure consists of two service primitives, as illustrated in Figure 3.

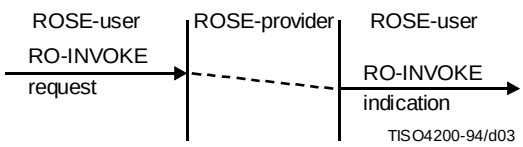


Figure 3 – RO-INVOKE service primitives

Table 2 lists the RO-INVOKE parameters.

Table 2 – RO-INVOKE parameters

Parameter name	Req.	Ind.
Operation-id	M	M(=)
Operation-type	U	
Argument	U	C(=)
Invoke-id	M	M(=)
Linked-id	U	C(=)
Priority	U	

8.1.1 Operation-id

This parameter identifies the operation to be performed as part of some operation contract. This parameter conveys the **&operationCode** from the operation definition.

8.1.2 Operation-type

This parameter classifies the operation as to:

– whether it is synchronous, or not.

The classification is derived from the **&synchronous** field of the operation definition. Absence of this parameter implies that the operation is asynchronous.

8.1.3 Argument

This parameter is the argument of the invoked operation. This parameter can be present if and only if the **&ArgumentType** field is present in the operation definition. If present, its type shall be as indicated by that field.

8.1.4 Invoke-id

This parameter identifies the request of a RO-INVOKE service and is used to correlate this request with the corresponding replies (RO-RESULT, RO-ERROR, RO-REJECT-U, and RO-REJECT-P services) or the invocation by the performer of linked child-operations (RO-INVOKE). This parameter has to be supplied by the requestor of the service.

This parameter distinguishes several requests of the service the requestor may have in progress (asynchronous operations). The requestor may begin to reuse Invoke-id values whenever it chooses, subject to the constraint that it may not reuse an Invoke-id value that was previously assigned to a request of the service for which it expects, but has not yet received, a reply or the invocation of a linked child-operation.

The ROSE-user to which an RO-INVOKE indication is issued, assumes that an Invoke-id value violating the above rule is a duplicate; and therefore, it does not perform the invoked operation. Instead, it rejects the duplicate invocation.

If the operation does not always report its outcome, the requestor of this service may reuse an Invoke-id value after a reasonably long period of time, or if the reply is carried by other means (e.g. by the result of a “have-you-finished” operation).

In some application contexts peer ROSE-users may communicate Invoke-id values. To support this, the set of allowed values of the Invoke-id parameter is used in (see Annex A of ITU-T Rec. X.880 | ISO/IEC 13712-1) defining the generic ROS PDUs.

8.1.5 Linked-id

If this parameter is present, the invoked operation is a child-operation and the parameter identifies the invocation of the linked parent-operation. This parameter has to be supplied by the requestor of the service. The value is that of the Invoke-id parameter of the RO-INVOKE indication primitive of the parent-operation.

8.1.6 Priority

This parameter defines the priority assigned to the transfer of the corresponding APDU with respect to the other APDUs to be exchanged between the AEs. The lower the value, the higher the priority. If several APDUs with the same priority are awaiting transfer, they are transferred “first in, first out”.

NOTES

1 The priority parameter is used only in conjunction with the RTSE as a transfer service. It shall not be used in any other realization.

2 The Priority parameter has an effect in the case of a two-way alternate association in that it prioritizes the sending of APDUs, and may be used to determine when to request the Turn to send APDUs. The Priority parameter may also have a local effect in the case of a two-way simultaneous association.

3 The Priority of a reply (RO-RESULT, RO-ERROR, and RO-REJECT-U) should normally be higher (lower in value) than the priority of the corresponding invocation.

When present, the Priority must be in the range allowed for by the **&InvokePriority** field of the operation definition.

8.2 RO-RESULT service

The RO-RESULT service is used by a ROSE-user to reply to a previous RO-INVOKE indication in the case of a successfully performed operation. This service is a non-confirmed service.

The related service structure consists of two service-primitives, as illustrated in Figure 4.

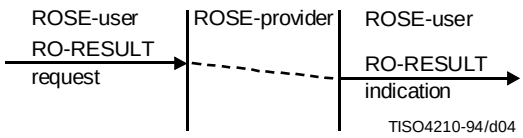


Figure 4 – RO-RESULT service primitives

Table 3 lists the RO-RESULT service parameters.

Table 3 – RO-RESULT parameters

Parameter name	Req.	Ind.
Operation-id	U	C(=)
Result	U	C(=)
Invoke-id	M	M(=)
Priority	U	

8.2.1 Operation-id

This parameter identifies the operation whose result is being reported after being performed as part of some operation contract. This parameter conveys the **&operationCode** from the operation definition. This parameter shall be present only if the Result parameter is present.

8.2.2 Result

This parameter is the result of the operation. This parameter may be present if and only if the **&ResultType** field is present in the operation definition. If present, its type shall be as indicated by that field.

8.2.3 Invoke-id

This parameter identifies the corresponding invocation (see 8.1.4). This parameter has to be supplied by the requestor of the service. The value is that of the corresponding RO-INVOKE indication primitive.

8.2.4 Priority

This parameter defines the priority assigned to the transfer of the corresponding APDU (see 8.1.6).

NOTE – The priority parameter is used only in conjunction with the RTSE as a transfer service. It shall not be used in any other realization.

When present, Priority must be in the range allowed for by the **&ResultPriority** field of the operation definition.

8.3 RO-ERROR service

The RO-ERROR service is used by a ROSE-user to reply to a previous RO-INVOKE indication in the case of an unsuccessfully performed operation. This service is a non-confirmed service. The related service structure consists of two service-primitives as illustrated in Figure 5.

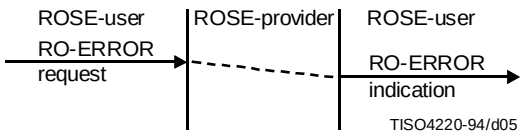


Figure 5 – RO-ERROR service primitives

Table 4 lists the RO-ERROR service parameters.

Table 4 – RO-ERROR parameters

Parameter name	Req.	Ind.
Error-id	M	M(=)
Parameter	U	C(=)
Invoke-id	M	M(=)
Priority	U	

8.3.1 Error-id

This parameter identifies the error being reported due to the inability to perform the invoked operation. This parameter conveys the **&errorCode** from the error definition.

8.3.2 Parameter

This is the parameter of the error, and may be present if and only if the **&ParameterType** field is present in the error definition. If present, its type shall be as indicated by that field.

8.3.3 Invoke-id

This parameter identifies the corresponding invocation (see 8.1.4). This parameter has to be supplied by the requestor of the service. The value is that of the corresponding RO-INVOKE indication primitive.

8.3.4 Priority

This parameter defines the priority assigned to the transfer of the corresponding APDU (see 8.1.6).

NOTE – The priority parameter is used only in conjunction with the RTSE as a transfer service. It shall not be used in any other realization.

When present, the Priority must be in the range allowed for by the **&ErrorPriority** field of the error definition.

8.4 RO-REJECT-U service

The RO-REJECT-U service is used by a ROSE-user to reject a request (RO-INVOKE indication) of the other ROSE-user if it has detected a problem. The RO-REJECT-U service may also be used by a ROSE-user to reject a reply (RO-RESULT indication, RO-ERROR indication) from the other ROSE-user. However, to avoid violating the sequencing rules of other ASEs in some application contexts, a ROSE-user may choose not to use the RO-REJECT-U service to reject replies. This service is a non-confirmed service.

The related service structure consists of two service-primitives, as illustrated in Figure 6.

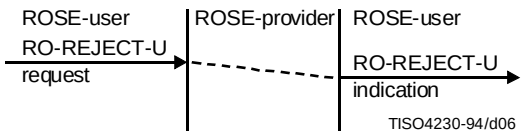


Figure 6 – RO-REJECT-U service primitives

Table 5 lists the RO-REJECT-U service parameters.

Table 5 – RO-REJECT-U parameters

Parameter name	Req.	Ind.
Reject-reason	M	M(=)
Invoke-id	M	M(=)
Priority	U	

8.4.1 Reject-reason

This parameter specifies the reason for rejection as follows:

a) **Invoke-problem**: user-reject of an RO-INVOKE indication primitive with values:

- **duplicate-invocation** – Signifies that the Invoke-id parameter violates the assignment rules of 8.1.4 [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.3.3 a)];
- **unrecognized-operation** – Signifies that the operation is not one of those agreed between the ROSE-users [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.3.3 b)];
- **mistyped-argument** – Signifies that the type of the operation argument supplied is not that agreed between the ROSE-users [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.3.3 d)];
- **resource-limitation** – The intended performer is not willing to perform the invoked operation due to resource limitation;
- **release-in-progress** – The intended performer is not willing to perform the invoked operation because it is about to release the application-association;
- **unrecognized-linked-ID** – Signifies that there is no operation in progress with an Invoke-id equal to the specified Linked-id for which no outcome has been reported [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.3.3 b)];
- **linked-response-unexpected** – Signifies that the invoked operation referred to by the Linked-id is not one which allows linked-operations to be invoked in response [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.3.3 b)];
- **unexpected-linked-operation** – Signifies that the invoked child-operation is not one that is allowed by the invoked parent-operation referred to by the Linked-id. [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.3.3 b)];

b) **Return-result-problem**: user-reject of an RO-RESULT indication primitive with values:

- **unrecognized-invocation** – Signifies that no operation with the specified Invoke-id is in progress for which a report of the outcome is expected [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.4.3 a)];
- **result-response-unexpected** – Signifies that the invoked operation does not report a result [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.4.3 a)];
- **mistyped-result** – Signifies that the type of the Result parameter supplied is not that agreed between the ROSE-users [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.4.3 b)];

c) **Return-error-problem**: user-reject of an RO-ERROR indication primitive with values:

- **unrecognized-invocation** – Signifies that no operation with the specified Invoke-id is in progress for which the report of the outcome is expected [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.5.3 a)];
- **unrecognized-error** – Signifies that the reported error is not one of those agreed between the ROSE-users [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.5.3 b)];
- **unexpected-error** – Signifies that the reported error is not one that the invoked operation may report [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.5.3 b)];
- **mistyped-parameter** – Signifies that the type of the error parameter supplied is not that agreed between the ROSE-user [see also ITU-T Rec. X.880 | ISO/IEC 13712-1, 9.5.3 c)].

This parameter has to be supplied by the requestor of the service.

8.4.2 Invoke-id

This parameter identifies the corresponding invocation (see 8.1.4). This parameter has to be supplied by the requestor of the service. The value is that of the rejected RO-INVOKE indication, RO-RESULT indication, or RO-ERROR indication primitive.

8.4.3 Priority

This parameter defines the priority assigned to the transfer of the corresponding APDU (see 8.1.6, 8.2.4 and 8.3.4).

NOTE – The priority parameter is used only in conjunction with the RTSE as a transfer service. It shall not be used in any other realization.

8.5 RO-REJECT-P service

The RO-REJECT-P service is used to advise a ROSE-user of a corrupted ROSE APDU detected by the ROSE service provider. This service is a provider-initiated service. The RO-REJECT-P indication primitive arises as a result of an earlier RO-INVOKE, RO-RESULT, RO-ERROR or RO-REJECT – U request primitive issued by the same RO-user.

The RO-REJECT-P service can also be used to inform a ROSE-user of an unsuccessful transfer of a ROSE APDU by the ROSE service provider due to a problem in the underlying layers (e.g. association abort).

The related service structure consists of a single service-primitive, as illustrated in the lower portion of Figure 7.

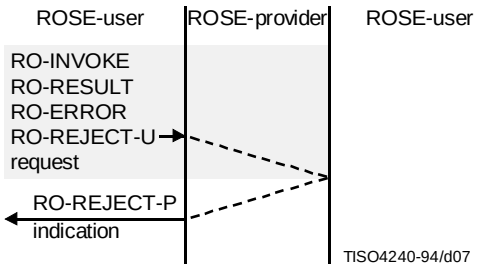


Figure 7 – RO-REJECT-P service primitive

Table 6 lists the RO-P-REJECT service parameters.

Table 6 – RO-REJECT-P parameters

Parameter name	Ind.
Invoke-id	O
Reject-reason	O

8.5.1 Invoke-id

This parameter identifies the corresponding ROS APDU (see 8.1.4) that has been rejected at the remote end. This parameter is supplied by the ROSE-provider. The value is that of the rejected RO-INVOKE request, RO-RESULT request, RO-ERROR request or RO-REJECT-U request primitives. This parameter may be replaced by the value **NULL** if an Invoke-id could not be derived by the ROSE service-provider.

8.5.2 Reject-reason

This parameter specifies the reason for rejection as follows:

a) **General-problem**: provider-reject of an APDU with values:

– **unrecognized-APDU** – Signifies that the type of the APDU, as evidenced by its tag, is not one of the four generic ROS PDUs defined in ITU-T Rec. X.880 | ISO/IEC 13712-1;

– **mistyped-APDU** – Signifies that the structure of the APDU does not conform to that defined in ITU-T Rec. X.880 | ISO/IEC 13712-1;

– **badly-structured-APDU** – Signifies that the structure of the APDU cannot be determined based on the negotiated presentation context.

This parameter is supplied by the ROSE service-provider.

8.6 RO-BIND service

The RO-BIND service is available only when the association contract being realized includes a connection package. It enables one ROSE-user to invoke and the other to perform the **&bind** operation of that connection package, in order to dynamically establish an association. This service is a confirmed service.

The related service structure consists of four service primitives, as illustrated in Figure 8.

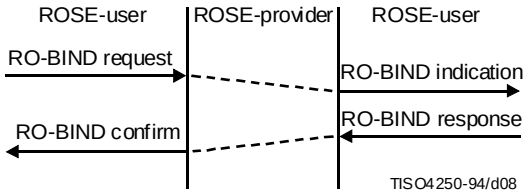


Figure 8 – RO-BIND service primitives

Table 7 lists the RO-BIND parameters.

Table 7 – RO-BIND Parameters

Parameter name	Req.	Ind.	Resp.	Conf.
Application context name	M	M(=)		
Operation-type	M			
Argument	C	C(=)		
Unbind can fail		M	C	C(=)
Outcome			M	M(=)
Bind-result			C	C(=)
Bind-error parameter			C	C(=)

8.6.1 Application context name

This parameter identifies the ROSE-based application context applying to the association being established. It takes the value of the **&id** field of that context.

NOTES

1 The information on whether the "responder unbind available" is derived from the **&associationRealization** and **&transferRealization** fields chosen in the application context definition identified by this parameter.

2 If the RTSE is chosen as the association and transfer realization, the responder cannot unbind.

8.6.2 Operation-type

See 8.1.2. For the present purpose, only the classification as to whether or not the operation is synchronous is significant.

8.6.3 Argument

This parameter is the argument of the invoked bind operation. This parameter may be present if and only if the **&ArgumentType** field is present in the operation definition. If present, its type shall be as indicated by that field.

8.6.4 Unbind can fail

This parameter takes the values "true" and "false". If the **&unbindCanFail** field of the relevant connection package is **FALSE** then the value of the parameter shall be "false" on all primitives. If the value on the indication primitive is "false" then it shall also be "false" on the response primitive. The setting of this parameter on the response and confirm primitives indicates whether or not the outcome "error-bound" can be issued in an RO-UNBIND response. This parameter is present on the response or confirm primitives if and only if the "outcome" parameter has the value "result".

NOTE – The possibility that the "unbind can fail" implies the use of the negotiated release functional unit of the Session service.

8.6.5 Outcome

This parameter indicates the outcome of the bind invocation, taking either the symbolic value "result" or "error".

8.6.6 Bind-result

This parameter may be present if and only if the Outcome parameter takes the value "result", and the **&ResultType** field is present in the operation definition. If present, its type shall be as indicated by that field.

8.6.7 Bind-error parameter

This parameter may be present if and only if the Outcome parameter takes the value "error", and the **&ParameterType** field is present in the error definition. If present, its type shall be as indicated by that field.

8.7 RO-UNBIND service

The RO-UNBIND service is available only when the association contract being realized includes a connection package. It enables one ROSE-user to invoke and the other to perform the **&unbind** operation of that connection package, in order to release a dynamically established association. If the **&responderCanUnbind** field of the package is **FALSE**, only the association initiator can invoke the unbind operation. This service is a confirmed service.

The related service structure consists of four service primitives, as illustrated in Figure 9.

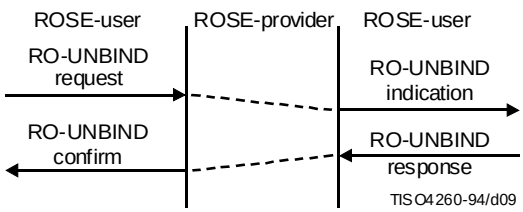


Figure 9 – RO-UNBIND service primitives

Table 8 lists the RO-UNBIND parameters.

Table 8 – RO-UNBIND Parameters

Parameter name	Req.	Ind.	Resp.	Conf.
Argument	C	C(=)		
Outcome			M	M(=)
Unbind-result			C	C(=)
Unbind-error parameter			C	C(=)

8.7.1 Argument

This parameter is the argument of the invoked unbind operation. This parameter may be present if and only if the **&ArgumentType** field is present in the operation definition. If present, its type shall be as indicated by that field.

8.7.2 Outcome

This parameter indicates the outcome of the unbind invocation, one of the symbolic values “result”, “error-bound” or “error-unbound”. If the “error-bound” outcome arises then the association remains bound despite the attempt to unbind.

8.7.3 Unbind-result

This parameter may be present if and only if the Outcome parameter takes the value “result”, and the **&ResultType** field is present in the operation definition. If present, its type shall be as indicated by that field.

8.7.4 Unbind-error parameter

This parameter may be present if and only if the Outcome parameter takes an “error” value, and the **&ParameterType** field is present in the error definition. If present, its type shall be as indicated by that field.

9 Sequencing information

This clause defines the constraints on the use of ROSE services.

9.1 Associations

9.1.1 Figure 10 shows the outer sequencing for an association where the association contract does not include a connection package. The remaining figures cover the sequencing rules where a connection package is involved. Figure 11 shows the permitted sequences of events for the association initiator, while Figure 12 shows the substructure of the “unbind pending” state. Figures 13 and 14 show the analogous sequences for the association responder.

9.1.2 The permitted sequences for “OPERATE” are as described in 9.2.

Legend for Figures 10-14

↓	Request or response
↑	Indication or confirm
Start	A transfer service is made available
Stop	A transfer service is no longer available
(UN)BIND	RO-(UN)BIND
(UB)BIND+	RO-(UN)BIND; outcome “result”
BIND–	RO-BIND; outcome “error”
UNBIND-ub	RO-UNBIND; outcome “error-unbound”
UNBIND-b	RO-UNBIND; outcome “error-bound”
ABORT	unnegotiated termination of the association
OPERATE	RO-INVOKE, RO-RESULT, or RO-ERROR for normal operation or error RO-REJECT-U, RO-REJECT-P

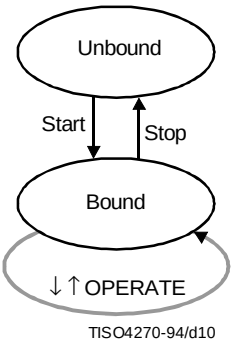


Figure 10 – Association contract without connection package

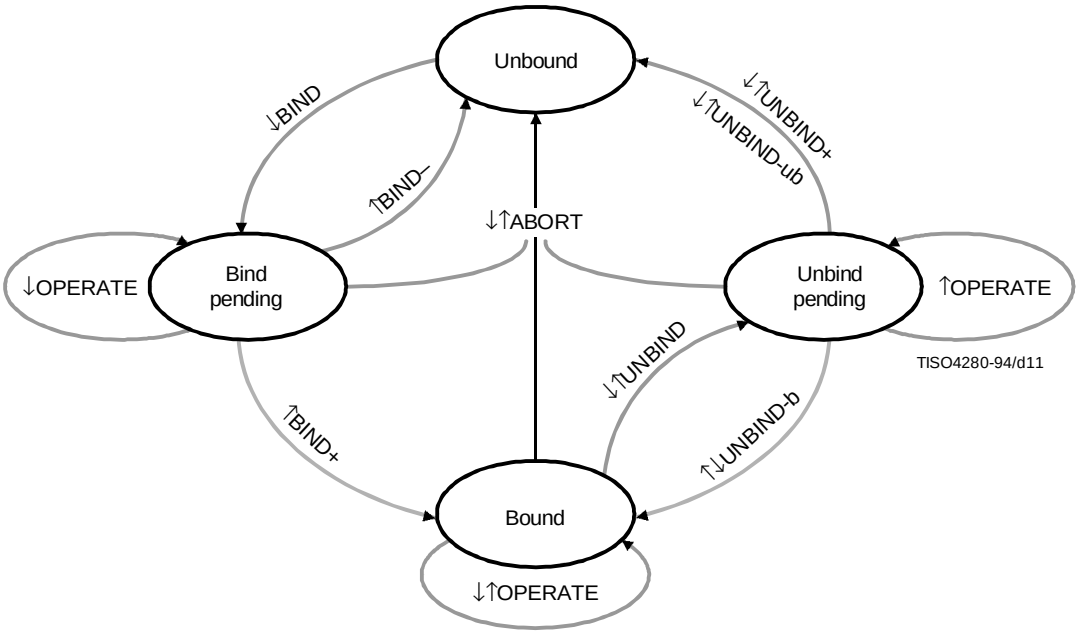


Figure 11 – Permitted sequences for association initiator

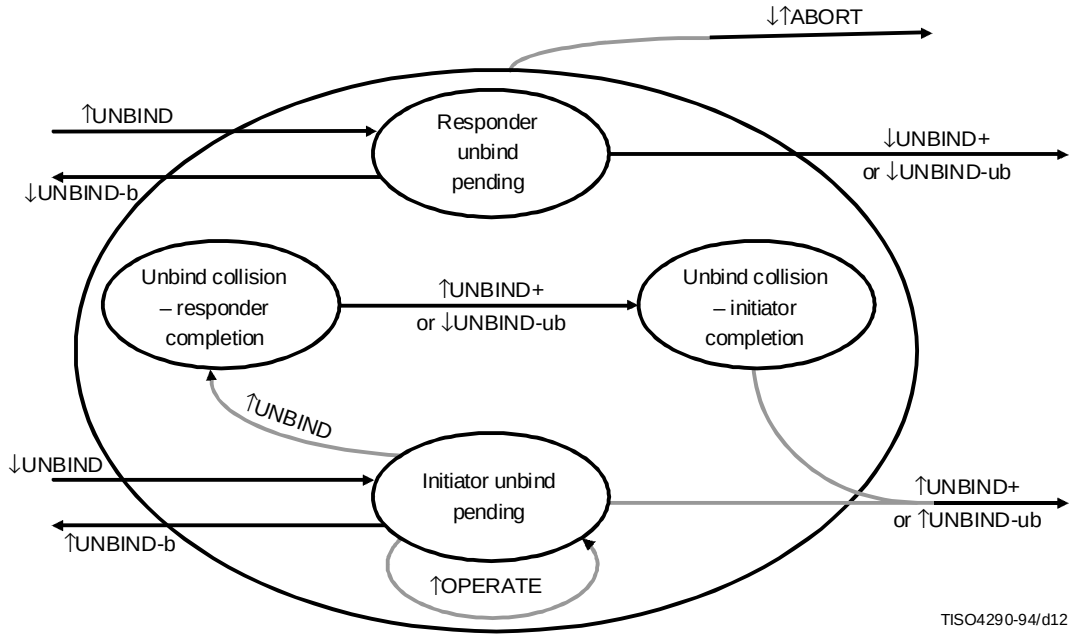


Figure 12 – Substructure of "unbind pending" for association initiator

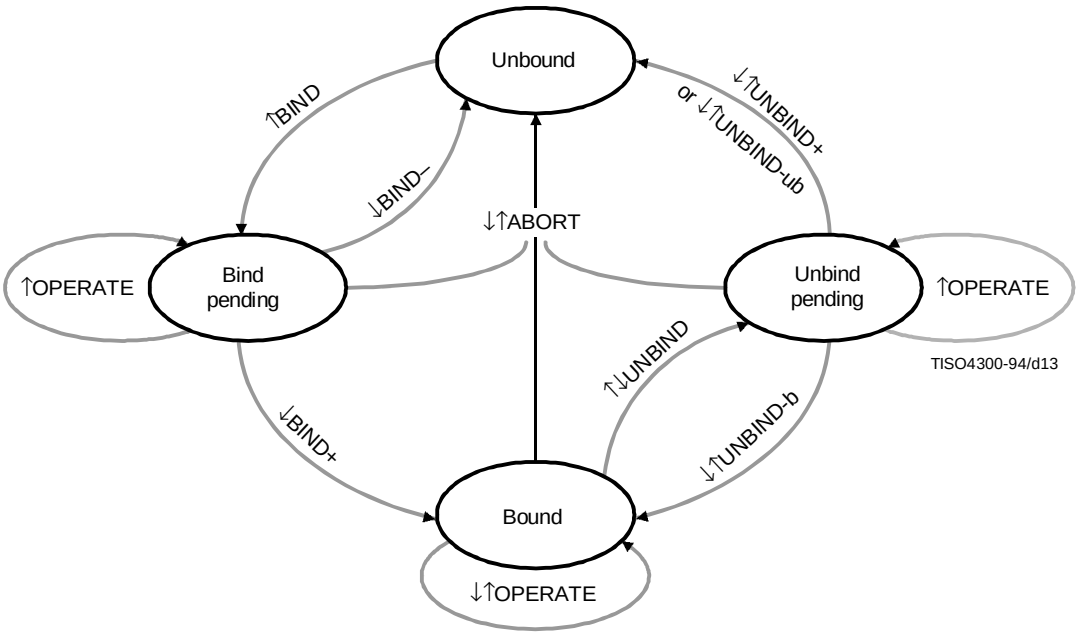


Figure 13 – Permitted sequences for association responder

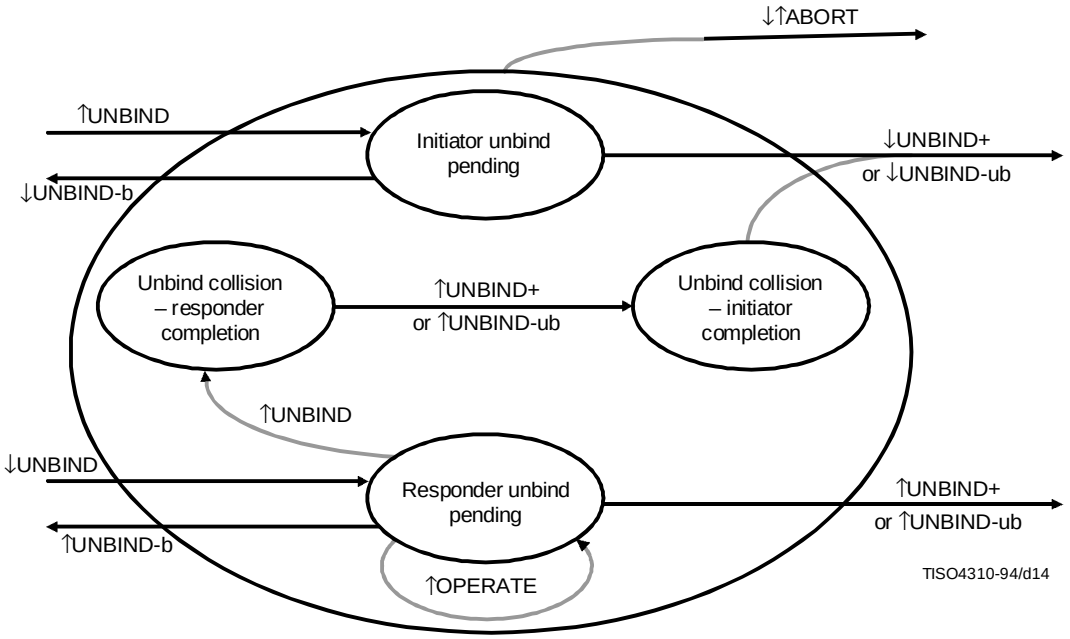


Figure 14 – Substructure of "unbind pending" for association responder

9.2 Operations

9.2.1 A series of primitives between an operation-specific ASE and ROSE may be associated with a single operation invocation. From the viewpoint of its invoker, such a series consists of an RO-INVOKE request together with:

– (optionally) an RO-RESULT, RO-ERROR, RO-REJECT-U, or RO-REJECT-P indication whose Invoke-id value is the same as that of the RO-INVOKE; and

– (optionally) one or more RO-INVOKE indications whose Linked-id value is the same as the Invoke-id value of the RO-INVOKE.

NOTE – A linked invoke is thus related to two of these series, and must obey any constraints imposed through either.

9.2.2 Figure 15 shows the constraints on the order in which such primitives may occur. The action of "forget" takes place as a local matter but not before it is adjudged that the series is complete.

9.2.3 Figure 16 shows the corresponding constraints on the primitives which take place at the performer. Here the "forget" action takes place after the performance of the operation is complete, and when it is known that the return will not have to be re-sent. An RO-INVOKE indication bearing the same Invoke-id as one which has finished is treated as a new invocation, while if it bears the Invoke-id of one which is running this is rejected as a duplicate invocation.

9.3 Further sequencing rules

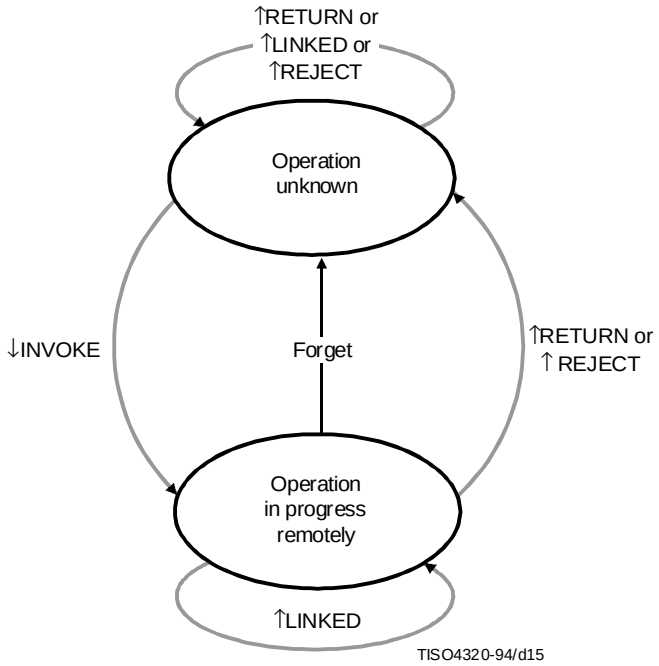
9.3.1 All primitives within a series as described in 9.2.1 shall occur within the same association.

9.3.2 A series of primitives as described in 9.2.1 and for which the operation identified by the RO-INVOKE request is synchronous shall not overlap in time with another such series.

9.3.3 "OPERATE" events associated with the invocation of synchronous operations are possible in the "bind pending" state only if the bind operation is not synchronous.

NOTE – It can be deduced from the requirement of 9.3.1 that such events can only be RO-INVOKE requests (for the initiator) or indications (for the responder).

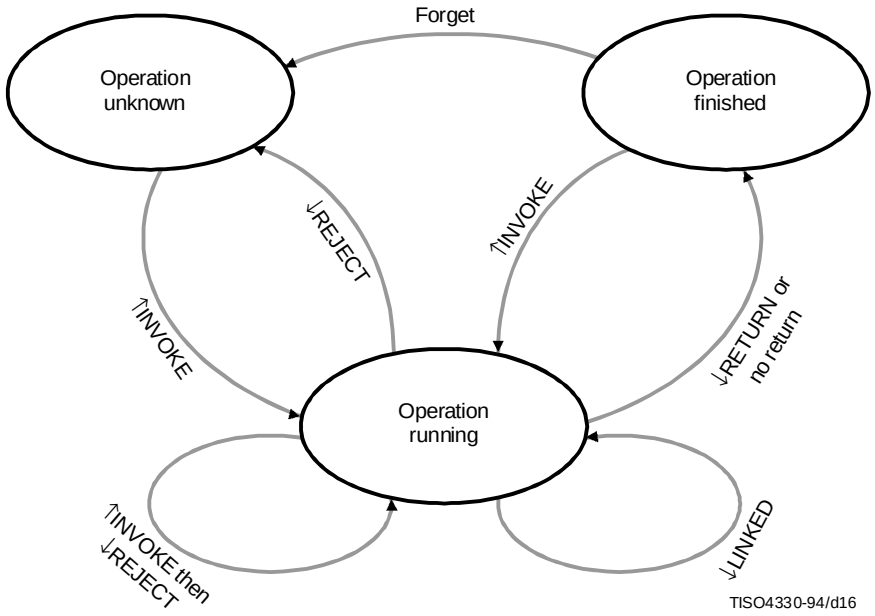
9.3.4 "OPERATE" events associated with the invocation of synchronous operations are possible in the "unbind pending" state only if the unbind operation is not synchronous.



↓	Request
↑	Indication
INVOKE	RO-INVOKE
RETURN	RO-RESULT or RO-ERROR
LINKED	RO-INVOKE for linked operation
REJECT	RO-REJECT-U or RO-REJECT-P

NOTE – The receipt of a ↑ RETURN, ↑ LINKED or a ↑ REJECT in the "operation unknown" state does not cause a state transition but may be viewed as a protocol error on the part of the remote end. This may lead to a rejection in the case of a ↑ RETURN or a ↑ LINKED.

Figure 15 – Permitted sequences for operation invoker



NOTE – The duration of the "operation finished" state is implementation dependent.

Figure 16 – Permitted sequences for operation performer

9.4 Invoke-id management

9.4.1 The invoker of an operation shall choose, for the invoke-id to be used in the RO-INVOKE request, an integer which is not already in use for an operation in progress remotely on that association (see Figure 15).

9.4.2 It is recommended that an invoke-id value not be reused immediately on the operation invocation which resulted in the previous use becoming unknown (see Figure 15). Instead, a further interval should be allowed to elapse, to ensure that no remnants of the previous use survive, either at the performer or in the medium.

NOTE – One approach is for each ROSE-user to use invoke-ids cyclically from a large subspace of the non-negative integers, the size of the subspace being chosen as a function of the invocation rate and the maximum invocation duration.

10 Mapping onto ROSE services

The invocation and performance of an operation *OP* in an association contract *C* (which we will label *C-OP*) can be viewed in terms of the sequence of service primitives illustrated in Figure 17.

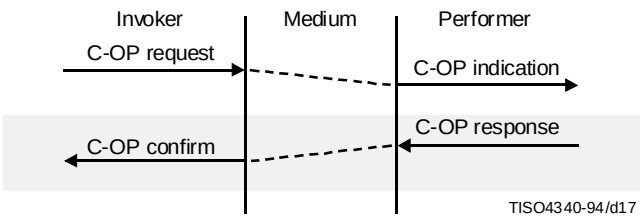


Figure 17 – Time sequence for operation invocation

The response and confirm only occur if and when *C-OP* reports its outcome. These primitives have a parameter "outcome" taking the value "result" or "error".

To realize the *C-OP* request and indication, the operation-specific ASE in the invoker's application entity issues an RO-INVOKE request to ROSE, its parameters citing the particular operation of interest. As a consequence, the ROSE in the performer's application-entity issues an RO-INVOKE indication to the appropriate operation-specific ASE. This is illustrated in Figure 18.

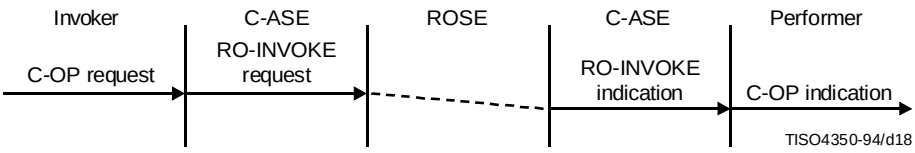


Figure 18 – Realization of operation invocation

If the operation was successfully performed, and its outcome is to be returned, the *C-OP* response and confirm is realized by the operation-specific ASE in the performer's application-entity issuing an RO-RESULT request to ROSE, its parameters describing the result. As a consequence, the ROSE in the invoker's application-entity issues an RO-RESULT indication to the appropriate operation-specific ASE. This is illustrated in Figure 19.

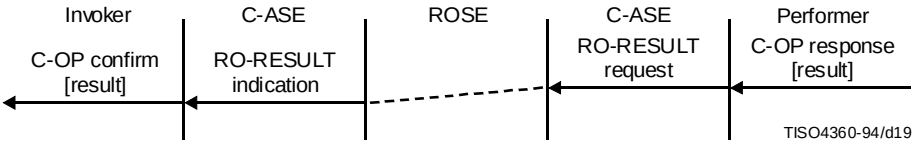


Figure 19 – Realization of performer reporting result

If the operation was unsuccessfully performed, and the outcome is to be returned, the *C-OP* response and confirm is realized by the operation-specific ASE in the performer's application-entity issuing an RO-ERROR request to ROSE, its parameters describing the error As a consequence, the ROSE in the invoker's application-entity issues an RO-ERROR indication to the appropriate operation-specific ASE. This is illustrated in Figure 20.

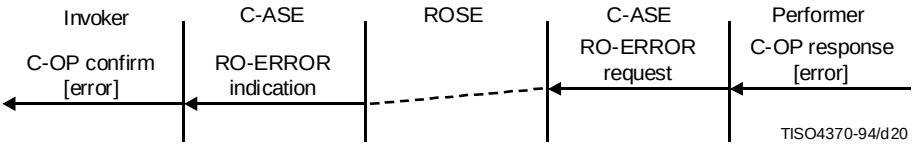


Figure 20 – Realization of performer reporting error

11 Mapping onto RO-BIND and RO-UNBIND services

The association-initiator establishes an application association through the invocation of the bind operation defined in an association contract C. This can be viewed in terms of the service primitives (which we shall label C-OPEN) shown in Figure 21.

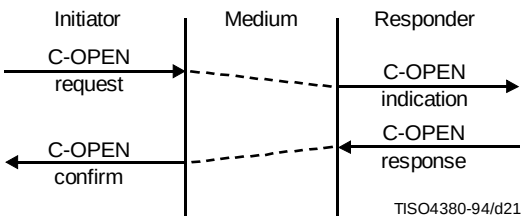


Figure 21 – Time sequence for association establishment

A similar figure can be used to show the release of an application association through the invocation of the unbind operation in the association contract. (This is invoked via the C-CLOSE services). However, depending on the specification of the connection package used in the association contract, only the association-initiator or either side can invoke the unbind (i.e. the C-CLOSE service.).

The bind operation is not used if there exists an established association with the appropriate characteristics required by the application context (which is specified as a parameter in the C-OPEN service). A successfully performed bind (respectively, unbind) operation establishes (respectively, releases) an application association. In an OSI realization, the bind operation and the unbind operation are mapped either on the ACSE services, or the RTSE services.

11.1 Mapping onto ACSE services

The bind operation is mapped on the A-ASSOCIATE services and the unbind operation mapped on the A-RELEASE service.

11.1.1 Mapping of a Bind operation

A bind operation is mapped on the A-ASSOCIATE service. The invocation of a bind operation is mapped on the A-ASSOCIATE request and A-ASSOCIATE indication service primitives. The argument value of the bind operation is mapped on the User Information parameter of the A-ASSOCIATE request service primitive. This is illustrated in Figure 22.

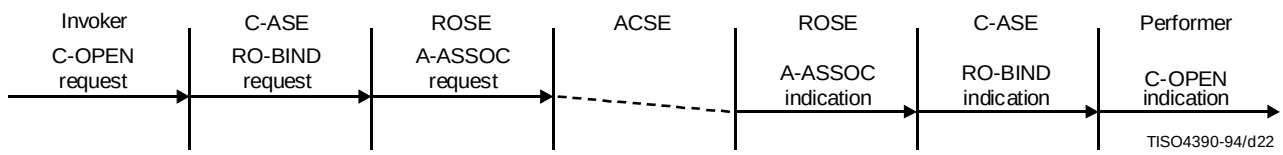


Figure 22 – Realization of a bind request using ACSE services

The reply of a bind operation is mapped on the A-ASSOCIATE response and A-ASSOCIATE confirm service primitives. This is shown in Figure 23.

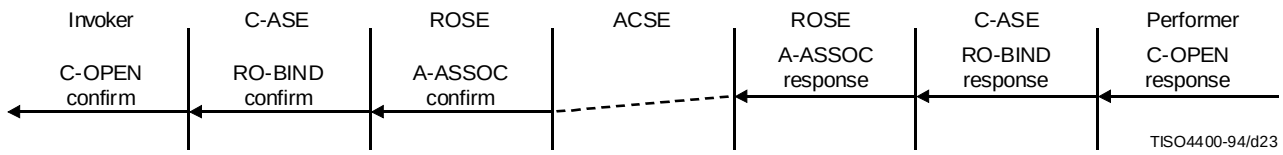


Figure 23 – Realization of a bind result using ACSE services

If the bind operation was successfully performed, the Result parameter of the A-ASSOCIATE service primitives is set to “accepted”, and the result value of the bind operation is mapped on the User Information parameter of these service primitives. If the bind operation was not successfully performed, the Result parameter value of the A-ASSOCIATE service primitives is set to “rejected (permanent)” or “rejected (transient)”, and the error value of the bind operation is mapped on the User Information parameter of these service primitives.

11.1.2 Mapping of a Unbind operation

An unbind operation is mapped on the A-RELEASE service. The invocation of an unbind operation is mapped on the A-RELEASE request and the A-RELEASE indication service primitives. The argument value of the unbind operation is mapped on the User Information parameter of the A-RELEASE service primitives. The Reason parameter value of the service primitives is set to “normal”. This is shown in Figure 24.

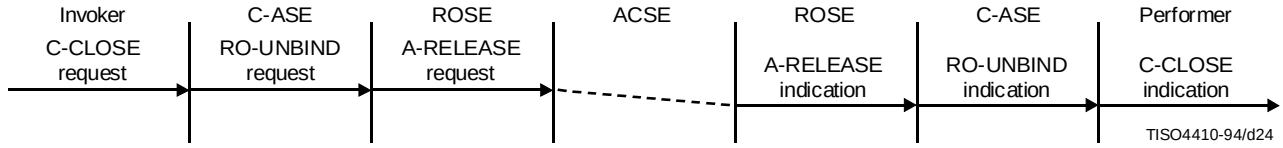


Figure 24 – Realization of an unbind using ACSE services

The reply of an unbind operation is mapped on the A-RELEASE response and A-RELEASE confirm service primitives. This is shown in Figure 25.

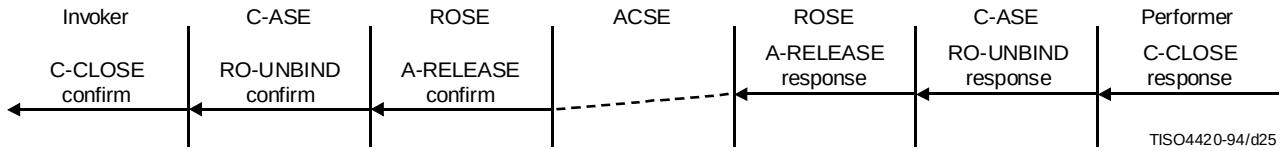


Figure 25 – Realization of an unbind result using ACSE services

If the unbind operation was successfully performed, the Reason parameter value of the A-RELEASE service primitives is set to “normal”, the result value of the unbind operation is mapped on the User Information parameter of the A-RELEASE service primitives, and the Result parameter of these service primitives is set to “affirmative”.

If the unbind operation was not successfully performed, the Reason parameter value of the A-RELEASE service primitives is set to “not finished”, the error value of the unbind operation is mapped on the User Information parameter of these service primitives, and the Result parameter of these service primitives is set to “negative”.

11.2 Mapping onto RTSE services

In another possible OSI realization, the bind operation is mapped on the RT-OPEN service and the unbind operation mapped on the RT-CLOSE service.

11.2.1 Mapping of a Bind operation

A bind operation is mapped on the RT-OPEN service. The invocation of a bind operation is mapped on the RT-OPEN request and RT-OPEN indication service primitives. This is shown in Figure 26.

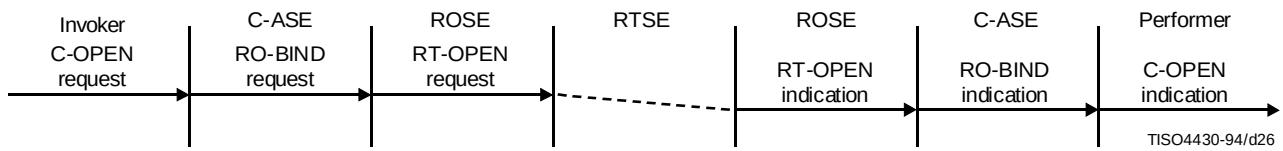


Figure 26 – Realization of a bind request using RTSE services

The argument value of the bind operation is mapped on the User-data parameter of these service primitives while the Dialogue-mode parameter value is set to “two-way-alternate”.

The reply of a bind operation is mapped on the RT-OPEN response and RT-OPEN confirm service primitives. This is shown in Figure 27.

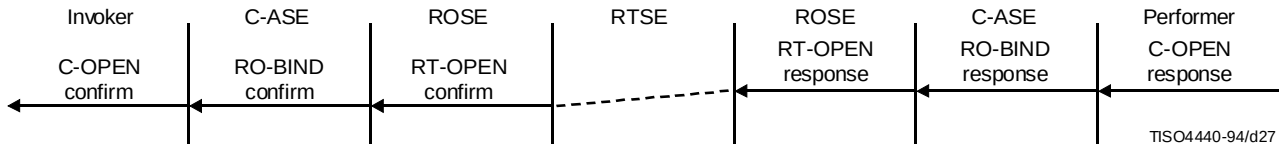


Figure 27 – Realization of a bind result using RTSE services

If the bind operation was successfully performed, the Result parameter of the RT-OPEN service primitives is set to “accepted”, and the result value of the bind operation is mapped on the User-data parameter of these service primitives. If the bind operation was not successfully performed, the Result parameter value of the RT-OPEN service primitives is set to “rejected (permanent)” or “rejected (transient)”, and the error value of the bind operation is mapped on the User data parameter of these service primitives.

11.2.2 Mapping of an Unbind operation

An unbind operation is mapped on the RT-CLOSE service. The invocation of an unbind operation is mapped on the RT-CLOSE request and the RT-CLOSE indication service primitives. This is shown in Figure 28.

Figure 28 – Realization of an unbind request using RTSE services

The argument value of the unbind operation is mapped on the User-data parameter of the RT-CLOSE service primitives. The Reason parameter value of these service primitives is set to “normal”.

The reply of an unbind operation is mapped on the RT-CLOSE response and RT-CLOSE confirm service primitives. This is shown in Figure 29.

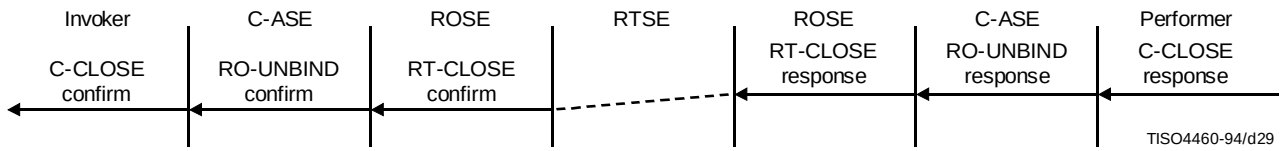


Figure 29 – Realization of an unbind result using RTSE services

If the unbind operation was successfully performed, the Reason parameter value of the RT-CLOSE service primitives is set to “normal”, while the result value of the unbind operation is mapped on the User-data parameter of these service primitives.

If the unbind operation was not successfully performed, the Reason parameter value of the RT-CLOSE service primitives is set to “not finished”, while the error value of the unbind operation is mapped on the User-data parameter of these service primitives.

Annex A

ASN.1 Modules

(This annex forms an integral part of this Recommendation | International Standard)

```
Remote-Operations-Information-Objects-extensions { joint-iso-itu-t remote-operations(4) informationObjects-
extensions(8) version1(0) }
DEFINITIONS ::=
BEGIN
-- exports everything
IMPORTS CONTRACT FROM Remote-Operations-Information-Objects { joint-iso-itu-t remote-operations(4)
informationObjects(5) version1(0) };


APPLICATION-CONTEXT ::= CLASS
{
        &associationContract            CONTRACT,
        &associationRealization         REALIZATION OPTIONAL,
        &transferRealization            REALIZATION,
        &AbstractSyntaxes               ABSTRACT-SYNTAX,
        &applicationContextName         OBJECT IDENTIFIER UNIQUE
}
WITH SYNTAX
{
        CONTRACT                        &associationContract
        [ ESTABLISHED BY                &associationRealization ]
        INFORMATION TRANSFER BY         &transferRealization
        ABSTRACT SYNTAXES               &AbstractSyntaxes
        APPLICATION CONTEXT NAME        &applicationContextName
}

REALIZATION ::= TYPE-IDENTIFIER

-- information objects ABSTRACT-SYNTAX and TYPE-IDENTIFIER are defined in ITU-T Rec. X.681 |
-- ISO/IEC 8824-2

END -- end of the information-objects-extensions module
```

Annex B

Guidelines for the use of the notation

(This annex does not form an integral part of this Recommendation | International Standard)

B.1 Example of information objects of class Application Context

Annex B of ITU-T Rec. X.880 | ISO/IEC 13712-1 provides an example of an association contract, **contract1,** which uses **connectionPackage1** for the explicit establishment of an association over which the association-initiator plays the role of the "consumer" when invoking the operations of (operation) **package1**. This subclause provides an application context which is a specific realization of **contract1**. This is defined as:

```
context1 APPLICATION-CONTEXT ::=
{
CONTRACT                          contract1
ESTABLISHED BY                    acse
INFORMATION TRANSFER BY           pData
ABSTRACT SYNTAXES                 { acse-object-identifier | package1-PCI }
APPLICATION CONTEXT NAME          objectIdentifierofApplicationContext1
}

-- where acse and pData are specified in ITU-T Rec. X.882 | ISO/IEC 13712-3 Annex C, and

package1-PCI ABSTRACT-SYNTAX ::=
{ ROS{ InvokeIdSet, AllOperations{ package1 }, AllOperations{ package1 } }
IDENTIFIED BY  objectIdentifierOfPackage1AbstractSyntax
}

acse-object-identifier OBJECT IDENTIFIER ::= { joint-iso-ccitt association-control(2) abstractSyntax(1) apdus(0)
version1(1) }
```

This application context, which is announced or negotiated by the identifier **objectIdentifierOfApplicationContext1**, uses the services of ACSE to establish the association using the bind operation defined as a part of the connection package in **contract1**. (See B.5 of ITU-T Rec. X.880 | ISO/IEC 13712-1.) After the association has been established, the P-DATA services are used for the transfer of each invocation or its response. During the association establishment phase, the abstract syntax in use is that of the PDUs of ACSE, while, during the data transfer phase, the abstract syntax is that given by **package1-PCI** which is, in effect, the ROS-PDUs parameterised (see 10.20 of ITU-T Rec. X.880 | ISO/IEC 13712-1) to report the invocation and result of all the operations in **package1**.

B.2 Releasing Application-associations in an Orderly Way

B.2.1 Introduction

ITU-T Rec. X.880 | ISO/IEC 13712-1 defines operations based upon, among other things, whether they always report their outcome and if they are synchronous. (This is determined by the presence of the fields **&alwaysReturns** and **&synchronous** in the operation information object class definition).

ITU-T Rec. X.880 | ISO/IEC 13712-1 also defines a connection package which is a specification of the roles of the two ROS-objects when dynamically establishing the association between them over which operations are invoked. Of particular relevance in the definition of a connection-package is whether the responder (as well as the initiator) of the association-establishment can unbind and whether the association can persist even after the attempt to unbind has failed. (These are indicated by the presence of the **&responderCanUnbind** and **&unbindCanFail** fields in the connection package information object class definition.)

In addition, ITU-T Rec. X.880 | ISO/IEC 13712-1 defines an association contract information object which specifies the roles (consumer or supplier or both) played by the association initiator and the association responder for each of the operation packages present in the contract. (These roles are indicated, respectively, by the presence of the fields **&InitiatorConsumerOf, &InitiatorSupplierOf** and **&OperationsOf** in the definition of the association contract information object class.)

By examining the definitions of the association contract and the operation packages contained therein, one can determine which operations (if any) are invoked by the initiator and the responder.

This annex defines the rules that ensure orderly release of an application association and the operations invoked over it.

B.2.2 Objectives

The rules in this annex are intended to achieve one of the following two objectives, depending upon the situation:

a) **The Exactly-Once Objective** – Ideally an application entity should be able to count on the invocation of an operation causing that operation to be performed exactly once, that is, not several times and not none at all.

b) **The At-Most-Once Objective** – In some circumstances the Exactly-Once Objective cannot be achieved. A lesser but still useful objective is that invoking an operation will cause that operation to be performed at most once, that is, perhaps not at all but never twice.

B.2.3 Definition of Rules

The following general rules apply in all circumstances:

G1 The performer shall report the result or error of each synchronous or asynchronous operation for which the **&alwaysReturns** field is set to **TRUE** over the same application-association over which the operation was invoked.

G2 The initiator (or the responder, if it is permitted to unbind) shall not invoke the RO-UNBIND service until the response has been received for each operation that it invoked for which a report of the outcome is expected.

The following specific rules apply in certain circumstances:

S1 Each time it exercises the RO-INVOKE service, the invoker shall supply a different Invoke-id even across a succession of application-associations. This enables the performer to achieve the At-Most-Once Objective by suppressing duplicates.

S2 If the performer encounters a duplicate Invoke-id in the RO-INVOKE service, it shall exercise the RO-REJECT-U service with duplicate-invocation as the Reject-reason. This helps to achieve the Exactly-Once Objective.

S3 The association-initiator (or the responder, if it can unbind) shall reject any invocations it has not performed before invoking the RO-UNBIND service.

S4 The association-initiator (or the responder, if it can unbind) shall respond (where a response is expected) to any invocations it has performed before invoking the RO-UNBIND service.

B.2.4 Application of Rules

The general rules always apply. The specific rules govern application-associations set up by association contracts with particular characteristics and the nature of the operations invoked over such associations.

a) **Application-associations where an examination of the association contract and the operation packages reveals that only the association initiator invokes operations** – In the case of synchronous and asynchronous operations whose outcome are always reported, specific rules S3 and S4 apply for the case when the responder can unbind. In the case of operations for which the outcome is sometimes reported, specific rules S1 and S2 apply if only the association-initiator can unbind while all the specific rules apply if the responder can unbind.

b) **Application-associations where an examination of the association contract and the contained operation packages reveals that only the association responder or both can invoke operations** – For synchronous and asynchronous operations whose outcome are always reported, specific rules S3 and S4 apply. (Any invocation issued by the association-responder (respectively association-initiator, if the responder can unbind) after the association-initiator (respectively, association-responder) has issued a release are lost. Response-rejects may be lost as well.) For operations whose outcome are sometimes reported, all the specific rules S1, S2, S3 and S4 apply.

For packages containing only synchronous or asynchronous operations which always report their outcome, the only constraint upon the values of the Invoke-ids that the invoker supplies to the RO-INVOKE service is that they be distinct over the life of the application-association.

Application-entities can make Invoke-ids unique to an invoker and across consecutive application-associations with the same intended performer by exchanging presentation-addresses at application-association establishment time and, for each presentation-address, making Invoke-ids integers that increase monotonically over some reasonable long period of time.

To ensure that an operation which sometimes reports its outcome has been performed exactly once, an application-entity shall elicit a Reject with the Reject-reason “duplicate-invocation” by invoking the operation twice (or more) with the same Invoke-id. Otherwise, the At-Most-Once Objective is all that is assured, suggesting that, on average, the invoker does not care whether a non-confirmed operation is performed.

B.2.5 Observations

Every synchronous or asynchronous operation that always reports its outcome is performed exactly once.

The usefulness of non-confirmed operations or conditionally confirmed operations (i.e. operations that sometimes report their outcomes) may depend on the specific application. Protocol designers may well be advised to define only operations that always report their outcome unless some very specialized requirements exists.

Annex C

Assignment of object identifier values

(This annex does not form an integral part of this Recommendation | International Standard)

The following object identifier value has been assigned in this Recommendation | International Standard:

Clause	Object Identifier Value
Annex A	**{ joint-iso-itu-t remote-operations(4) informationObjects-extensions(8) version1(0) }**