INTERNATIONAL TELECOMMUNICATION UNION

ITU-T X.229

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

OPEN SYSTEMS INTERCONNECTION

CONNECTION-MODE PROTOCOL SPECIFICATIONS

REMOTE OPERATIONS: PROTOCOL SPECIFICATION

ITU-T Recommendation X.229

(Extract from the *Blue Book*)

NOTES

1 ITU-T Recommendation X.229 was published in Fascicle VIII.5 of the *Blue Book*. This file is an extract from the *Blue Book*. While the presentation and layout of the text might be slightly different from the *Blue Book* version, the contents of the file are identical to the *Blue Book* version and copyright conditions remain unchanged (see below).

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

Recommendation X.229

REMOTE OPERATIONS: PROTOCOL SPECIFICATION[1)]

(Melbourne, 1988)

The CCITT,

considering

(a) that Recommendation X.200 defines the Basic Reference Model of Open Systems Interconnection (OSI) for CCITT Applications;

(b) that Recommendation X.210 defines the service conventions for describing the services of the OSI reference model;

(c) that Recommendation X.216 defines the Presentation Layer service;

(d) that Recommendation X.217 defines the Association Control service;

(e) that Recommendation X.218 defines the Reliable Transfer service;

(f) that Recommendation X.219 defines the Remote Operations service and notation;

(g) that there is a need for common Remote Operations support for various applications,

unanimously declares

that this Recommendation defines the Remote Operations protocol of Open Systems Interconnection for CCITT Applications as given in the Scope and Field of Application.

CONTENTS

---

1) Recommendation X.229 and ISO 9072-2 [Information processing systems - Text Communication-Remote Operations Part 2: Protocol specification] were developed in close collaboration and are technically aligned.

0 Introduction

This Recommendation specifies the protocol for the services provided by an application-service-element - the Remote Operations Service Element (ROSE) - to support interactive applications in a distributed open systems environment. This Recommendation is one of a set of Recommendations defining sets of application service-elements commonly used by a number of applications.

Interactions between entities of a distributed application are modelled as Remote Operations, and defined using a Remote Operations Notation. A Remote Operation is requested by one entity; the other entity attempts to perform the Remote Operation and then reports the outcome of the attempt. Remote Operations are supported by the ROSE.

This Recommendation is technically aligned with ISO 9072-2.

1 Scope and field of application

This Recommendation specifies the protocol (abstract syntax) and procedures for the Remote Operation Service Element (Recommendation X.219). The ROSE services are provided in conjunction with the Association Control Service Element (ACSE) services (Recommendation X.217) and the ACSE protocol (Recommendation X.227), optionally the Reliable Transfer Service Element (RTSE) services (Recommendation X.218) and the RTSE protocol (Recommendation X.228), and the presentation-service (Recommendation X.216).

The ROSE procedures are defined in terms of:

a) the interactions between peer ROSE protocol machines through the use of RTSE services or the presentation-service;

b) the interactions between the ROSE protocol machine and its service-user.

This Recommendation specifies conformance requirements for systems implementing these procedures.

2 References

Recommendation X.200 - Reference model of open systems interconnection for CCITT applications (see also ISO 7498).

Recommendation X.208 - Specification of abstract syntax notation (see also ISO 8824).

Recommendation X.209 - Specification of basic encoding rules for the abstract syntax notation (see also ISO 8825).

Recommendation X.210 - Open systems interconnection layer service definition conventions (see also ISO/TR 8509).

Recommendation X.216 - Presentation service definition for open systems interconnection for CCITT applications (see also ISO 8822).

Recommendation X.217 - Association control service definition for CCITT applications (see also ISO 8649).

Recommendation X.218 - Reliable transfer: model and service definition (see also ISO 9066-1).

Recommendation X.219 - Remote operations: model, notation and service definition (see also ISO 9072-1).

Recommendation X.227 - Association control protocol specification for CCITT applications (see also ISO 8650)

Recommendation X.228 - Reliable transfer: protocol specification (see also ISO 9066-2).

3 Definitions

3.1 *Reference model definitions*

This Recommendation is based on the concepts developed in Recommendation X.200 and makes use of the following terms defined in it:

a) application layer;

b) application-process;

c) application-entity;

d) application-service-element;

e) application-protocol-data-unit;

f) application-protocol-control-information;

g) presentation-service;

h) presentation-connection;

i) session-service;

j) session-connection;

k) transfer syntax; and

l) user-element.

3.2 *Service conventions definitions*

This Recommendation makes use of the following terms defined in Recommendation X.210:

a) service-provider;

b) service-user;

c) confirmed service;

d) non-confirmed service;

e) provider-initiated service;

f) primitive;

g) request (primitive);

h) indication (primitive);

i) response (primitive); and

j) confirm (primitive).

3.3 *Presentation service definitions*

This Recommendation makes use of the following terms defined in Recommendation X.216:

a) abstract syntax;

b) abstract syntax name;

c) presentation context.

3.4 *Association control definitions*

This Recommendation makes use of the following terms defined in Recommendation X.217:

a) application-association; association;

b) application context;

c) association control service element.

3.5 *Reliable transfer definitions*

This Recommendation makes use of the following terms defined in Recommendation X.218:

a) reliable transfer service element.

3.6 *ROSE service definitions*

This Recommendation makes use of the following terms defined in Recommendation X.219:

a) association-initiating-application-entity; association-initiator;

b) association-responding-application-entity; association-responder;

c) invoking-application-entity; invoker;

d) performing-application-entity; performer;

e) requestor;

f) acceptor;

g) linked-operations;

h) parent-operation;

i) child-operation;

j) RO-notation;

k) remote operation service element;

l) ROSE-provider;

m) ROSE-user;

n) RTSE-user;

o) remote operations.

3.7 *Remote operation protocol specification definitions*

For the purpose of this Recommendation the following definitions apply:

3.7.1 **remote-operation-protocol-machine:**

The protocol machine for the remote operation service element specified in this Recommendation.

3.7.2 **requesting-remote-operation-protocol-machine:**

The remote-operation-protocol-machine whose service-user is the requestor of a particular remote operation service element service.

3.7.3 **accepting-remote-operation-protocol-machine:**

The remote-operation-protocol-machine whose service-user is the acceptor for a particular remote operation service element service.

4 Abbreviations

4.1 *Data units*

APDU application-protocol-data-unit.

4.2 *Types of application-protocol-data-units*

The following abbreviations have been given to the application-protocol-data-units defined in this Recommendation.

ROIV	RO-INVOKE application-protocol-data-unit
RORS	RO-RESULT application-protocol-data-unit
ROER	RO-ERROR application-protocol-data-unit
RORJ	RO-REJECT application-protocol-data-unit

4.3 *Other abbreviations*

The following abbreviations are used in this Recommendation.

AE	application entiry
ACSE	association control service element
ASE	application service element
RO (or ROS)	remote operations
ROPM	remote operations protocol machine
ROSE	remote operations service element
RT	reliable transfer
RTSE	reliable transfer service element

5 Conventions

This Recommendation employs a tabular presentation of its APDU fields. In clause 7, tables are presented for each ROSE APDU. Each field is summarized using the following notation:

M	presence is mandatory
U	presence is a ROSE-user option
req	source is related request primitive
ind	sink is related indication primitive
resp	source is related response primitive
conf	sink is related confirm primitive
sp	source or sink is the ROPM

The structure of each ROSE APDU is specified in clause 9 using the abstract syntax notation of Recommendation X.208.

6 Overview of the protocol

6.1 *Service provision*

The protocol specified in Recommendation provides the ROSE services defined in Recommendation X.219. These services are listed in Table 1 /X.229.

TABLE 1/X.229

ROSE services summary

Service	Type
RO-INVOKE	Non-confirmed
RO-RESULT	Non-confirmed
RO-ERROR	Non-confirmed
RO-REJECT-U	Non-confirmed
RO-REJECT-P	Provider-initiated

6.2 *Use of services*

The ROSE protocol specified in this Recommendation needs a transfer service to pass information in the form of ROSE APDUs between peer application-entities (AEs).

Two transfer services may be used alternatively:

a) the RTSE services, if the RTSE is included in the application-context; or

b) the presentation-service, if the RTSE is not included in the application-context.

In both cases, an existing application-association, established and released by means of the ACSE services, is assumed.

6.2.1 *Use of the RTSE services*

If the RTSE is included in the application-context, this Recommendation assumes that the ROPM is the sole user of the RT-TRANSFER service and the RT-TURN-GIVE service.

The initiating AE may only request the release of the application-association by means of the RT-CLOSE service if it possesses the turn. Therefore the RTSE-user and the ROPM are the user of the RT-TURN-PLEASE service.

The ROPM is the user of the RT-U-ABORT and RT-P-ABORT services.

6.2.2 *Use of the presentation-service*

If the RTSE is not included in the application context, the ROPM is a user of the P-DATA service.

6.3 *Model*

The remote-operation-protocol-machine (ROPM) communicates with its service-user by means of primitives defined in Recommendation X.219. Each invocation of the ROPM controls a single application-association.

The ROPM is driven by ROSE service request primitives from its service-user, and by indication and confirm primitives of the RTSE services, or the presentation-service. The ROPM, in turn, issues indication primitives to its service-user, and request primitives on the used RTSE services, or the presentation-service. If the RTSE is included in the application-context, the RT-TRANSFER indication, RT-TRANSFER request and RT-TRANSFER confirm primitives are used. In the case of an application-context excluding RTSE, the presentation-service P-DATA request, and P-DATA indication primitives are used. In this case the transfer is not confirmed.

The reception of an ROSE service primitive, or of an RTSE service or of a presentation-service primitive, and the generation of dependent actions are considered to be individual.

During the exchange of APDUs, the existence of both, the association-initiating AE and the association-responding AE is presumed. How these AEs are created is beyond the scope of this Recommendation.

During the execution of operations, the existence of an application-association between the peer AEs is presumed. How this application-association is established and released is beyond the scope of this Recommendation (see Recommendations X.219, X.217, X.227, X.218 and X.228).

Note - Each application-association may be identified in an end system by an internal, implementation dependent mechanism so that the ROSE service-user and the ROPM can refer to it.

7 Elements of procedure

The ROSE protocol consists of the following elements of procedure:

a) invocation;

b) return-result;

c) return-error;

d) user-reject;

e) provider-reject.

In the following clauses, a summary of each of these elements of procedure is presented. This consists of a summary of the relevant APDUs, and high-level overview of the relationship between the ROSE service primitives, the APDUs involved, and the transfer service that is used.

The generic terms transfer service, transfer service-provider, transfer request, and transfer indication are used in the context of clause 7. Clause 8 describes how these generic service primitives are mapped either on to the RTSE services or the presentation-service.

In clause 9 a detailed specification of the ROSE APDUs is given using the notation defined in Recommendation X.208.

7.1 *Invocation*

7.1.1 *Purpose*

The invocation procedure is used by one AE (the invoker) to request an operation to be performed by the other AE (the performer).

7.1.2 *APDUs used*

The invocation procedure uses the RO-INVOKER (ROIV) APDU.

The fields of the ROIV APDU are listed in Table 2/X.229.

TABLE 2/X.229

ROIV APDU fields

Field name	Presence	Source	Sink
Invoke-ID	M	req	ind
Linked-ID	U	req	ind
Operation-value	M	req	ind
Argument	U	req	ind

7.1.3 *Invocation procedure*

This procedure is driven by the following events:

a) an RO-INVOKE request primitive from the requestor;

b) an ROIV APDU as user-data of a transfer indication primitive.

7.1.3.1 *RO-INVOKE request primitive*

The requesting ROPM forms an ROIV APDU from the parameter values of the RO-INVOKE request primitive. It issues a transfer request primitive. The user-data parameter of the transfer request primitive contains the ROIV APDU.

The requesting ROPM waits either for a transfer indication primitive from the transfer service-provider or any other primitive from the requestor.

7.1.3.2 *ROIV APDU*

The accepting ROPM receives an ROIV APDU from its peer as user-data on a transfer indication primitive. If any of the fields of the ROIV APDU are unacceptable to this ROPM, the provider-reject procedure is performed, and no RO-INVOKE indication primitive is issued by the ROPM.

If the ROIV APDU is acceptable to the accepting ROPM, it issues an RO-INVOKE indication primitive to the acceptor. The RO-INVOKE indication primitive parameters are derived from the ROIV APDU.

The accepting ROPM waits either for a transfer indication primitive from the transfer service-provider or any other primitive from the acceptor.

7.1.4 *Use of the ROIV APDU fields*

The ROIV fields are used as follows.

7.1.4.1 *Invoke-ID*

This is the Invoke-ID parameter value of the RO-INVOKE request primitive. lt appears as the Invoke-ID parameter value of the RO-INVOKE indication primitive.

The value of this field is transparent to the ROPM, however the value may be used in the provider reject procedure.

7.1.4.2 *Linked-ID*

This is the Linked-ID parameter value of the RO-INVOKE request primitive. It appears as the Linked-ID parameter value of the RO-INVOKE indication primitive.

The value of this field is transparent to the ROPM.

7.1.4.3 *Operation-value*

This is the Operation-value parameter value of the RO-INVOKE request primitive. lt appears as the Operation-value parameter value of the RO-INVOKE indication primitive.

The value of this field is transparent to the ROPM.

7.1.4.4 *Argument*

This is the Argument parameter value of the RO-INVOKE request primitive. It appears as the Argument parameter value of the RO-INVOKE indication primitive.

The value of this field is transparent to the ROPM.

7.2 *Return-result*

7.2.1 *Purpose*

The return-result procedure is used by one AE (the performer) to request the transfer of the result of a successfully performed operation to the other AE (the invoker).

7.2.2 *APDU's used*

The return-result procedure uses the RO-RESULT (RORS) APDU.

The fields of the RORS APDU are listed in Table 3/X.229.

TABLE 3/X.229

RORS APDU fields

Field name	Presence	Source	Sink
Invoke-ID	M	req	ind
Operation-value	U	req	ind
Result	U	req	ind

7.2.3 *Return-result procedure*

This procedure is driven by the following events:

a) an RO-RESULT request primitive from the requestor;

b) an RORS APDU are user-data of a transfer indication primitive.

7.2.3.1 *RO-RESULT request primitive*

The requesting ROPM forms an RORS APDU from the parameter values of the RO-RESULT request primitive. It issues a transfer request primitive. The user-data parameter of the transfer request primitive contains the RORS APDU.

The requesting ROPM waits either for a transfer indication primitive from the transfer service-provider or any other primitive from the requestor.

7.2.3.2 *RORS APDU*

The accepting ROPM receives an RORS APDU from its peer as user-data on a transfer indication primitive. If any of the fields of the RORS APDU are unacceptable to this ROPM, the provider-reject procedure is performed, and no RO-RESULT indication primitive is issued by the ROPM.

If the RORS APDU is acceptable to the accepting ROPM, it issues an RO-RESULT indication primitive to the acceptor. The RO-RESULT indication primitive parameters are derived from the RORS APDU.

The accepting ROPM waits either for a transfer primitive from the transfer service-provider or any other primitive from the acceptor.

7.2.4 *Use of the RORS APDU fields*

The RORS fields are used as follows.

7.2.4.1 *Invoke-ID*

This is the Invoke-ID parameter value of the RO-RESULT request primitive. lt appears as the Invoke-ID parameter value of the RO-RESULT indication primitive.

The value of this field is transparent to the ROPM, however the value may be used in the provider-reject procedure.

7.2.4.2 *Operation-value*

This is the Operation-value parameter value of the RO-RESULT request primitive. It appears as the Operation-value parameter value of the RO-RESULT indication primitive.

The value of this field is transparent to the ROPM.

This field shall be present only if the result field is present.

7.2.4.3 *Result*

This is the Result parameter value of the RO-RESULT request primitive. It appears as the Result parameter value of the RO-RESULT indication primitive.

The value of this field is transparent to the ROPM.

7.3 *Return-error*

7.3.1 *Purpose*

The return-error procedure is used by one AE (the performer) to request the transfer of the error information in the case of an unsuccessfully performed operation to the other AE (the invoker).

7.3.2 *APDUs used*

The return-error procedure uses the RO-ERROR (ROER) APDU.

The fields of the ROER APDU are listed in Table 4/X.229.

TABLE 4/X.229

ROER APDU fields

Field name	Presence	Source	Sink
Invoke-ID	M	req	ind
Error-value	M	req	ind
Error-parameter	U	req	ind

7.3.3 *Return-error procedure*

This procedure is driven by the following events:

a) an RO-ERROR request primitive from the requestor;

b) an ROER APDU as user-data of a transfer indication primitive.

7.3.3.1 *RO-ERROR request primitive*

The requesting ROPM forms an ROER APDU from the parameter values of the RO-ERROR request primitive. lt issues a transfer request primitive. The user-data parameter of the transfer request primitive contains the ROER APDU.

The requesting ROPM waits either for a transfer primitive from the transfer service-provider or any other primitive from the requestor.

7.3.3.2 *ROER APDU*

The accepting ROPM receives an ROER APDU from its peer as user-data on a transfer indication primitive. If any of the fields of the ROER APDU are unacceptable to this ROPM, the provider-reject procedure is performed, and no RO-ERROR indication primitive is issued by the ROPM.

If the ROER APDU is acceptable to the accepting ROPM, it issues an RO-ERROR indication primitive to the acceptor. The RO-ERROR indication primitive parameters are derived from the ROER APDU.

The accepting ROPM waits either for a transfer indication primitive from the transfer service-provider or any other primitive from the acceptor.

7.3.4 *Use of the ROER APDU fields*

The ROER fields are used as follows.

7.3.4.1 *Invoke-ID*

This is the Invoke-ID parameter value of the RO-ERROR request primitive. lt appears as the Invoke-ID parameter value of the RO-ERROR indication primitive.

The value of this field is transparent to the ROPM, however the value may be used in the provider-reject procedure.

7.3.4.2 *Error-value*

This is the Error-value parameter value of the RO-ERROR request primitive. lt appears as the Error-value parameter value of the RO-ERROR indication primitive.

The value of this field is transparent to the ROPM.

7.3.4.3 *Error-parameter*

This is the Error-parameter parameter value of the RO-ERROR request primitive. It appears as the Error-parameter parameter value of the RO-ERROR indication primitive.

The value of this field is transparent to the ROPM.

7.4 *User-reject*

7.4.1 *Purpose*

The user-reject procedure is used by one AE to reject the request (invocation) or reply (result or error) of the other AE.

7.4.2 *APDUs used*

The user-reject procedure uses the RO-REJECT (RORJ) APDU. This RORJ APDU is used in addition by the provider-reject procedure.

The fields of the RORJ APDU used for the user-reject procedure are listed in Table 5/X.229.

TABLE 5/X.229

RORJ APDU fields used for user-reject

Field name	Presence	Source	Sink
Invoke-ID	M	req	ind
Problem (choice of):	M	req	ind
Invoke-problem Return-result-problem Return-error-problem			

7.4.3 *User-reject procedure*

This procedure is driven by the following events:

a) an RO-REJECT-U request primitive from the requestor;

b) an RORJ APDU as user-data of a transfer indication primitive.

7.4.3.1 *RO-REJECT-U request primitive*

The requesting ROPM forms an RORJ APDU from the parameter values of the RO-REJECT-U request primitive. lt issues a transfer request primitive. The user-data parameter of the transfer request primitive contains the RORJ APDU.

The requesting ROPM waits either for a transfer indication primitive from the transfer service-provider or any other primitive from the acceptor.

7.4.3.2 *RORJ APDU*

The accepting ROPM receives an RORJ APDU from its peer as user-data on a transfer indication primitive. If any of the fields of the RORJ APDU are unacceptable to this ROPM, no RO-REJECT-U indication primitive is issued by the ROPM.

If the RORJ APDU is acceptable to the accepting ROPM and the fields of the RORJ APDU indicates a user reject (i.e. Invoke-problem, Return-result-problem, or Return-error-problem), it issues an RO-REJECT-U indication primitive to the acceptor. The RO-REJECT-U indication primitive parameters (Invoke-ID and Reject-reason) are derived from the RORJ APDU.

The accepting ROPM waits either for a transfer indication primitive from the transfer service-provider or any other primitive from the acceptor.

7.4.4 *Use of the RORJ APDU fields*

The RORJ fields are used as follows.

7.4.4.1 *Invoke-ID*

This is the Invoke-ID parameter value of the RO-REJECT-U request primitive. It appears as the Invoke-ID parameter value of the RO-REJECT-U indication primitive.

The value of this field is transparent to the ROPM.

7.4.4.2 *Problem*

This is the Problem parameter value of the RO-REJECT-U request primitive. lt appears as the Problem parameter value of the RO-REJECT-U indication primitive.

The values used by the user-reject procedure are:

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 Recommendation X.219.
- unrecognized-operation:
  signifies that the operation is not one of those agreed between the ROSE-users.
- mistyped-argument:
  signifies that the type of the operation argument supplied is not that agreed between the ROSE-users
- resource-limitation:
  the performing ROSE-user is not able to perform the invoked operation due to resource limitation.
- initiator-releasing:
  the association-initiator is not willing to perform the invoked operation because it is about to attempt 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.
- linked-response-unexpected:
  signifies that the invoked operation referred to by the Link-ID is not a parent-operation.
- unexpected-child-operation:
  signifies that the invoked child-operation is not one that the invoked parent-operation referred to by the Linked-ID allows.

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.
- result-response-unexpected:
  signifies that the invoked operation does not report a result.
- mistyped-result:
  signifies that the type of the Result parameter supplied is not that agreed between the ROSE-users.

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
- error-response-unexpected:
  signifies that the invoked operation does not report failure
- unrecognized-error:
  signifies that the reported error is not one of those agreed between the ROSE-users
- unexpected-error:
  signifies that the reported error is not one that the invoked operation may report
- mistyped-parameter:
  signifies that the type of the error parameter supplied is not that agreed between the ROSE-users.

7.5 *Provider-reject*

7.5.1 *Purpose*

The provider-reject procedure is used to inform the ROSE user and the peer ROPM, if an ROPM detects a problem.

7.5.2 *APDUs used*

The provider-reject procedure uses the RO-REJECT (RORJ) APDU. This RORJ APDU is used in addition by the user-reject procedure.

The fields of the RORJ APDU used for the provider-reject procedure are listed in Table 6/X.229.

TABLE 6/X.229

RORJ APDU fields used for provider-reject

Field name	Presence	Source	Sink
Invoke-ID	M	sp	ind
Problem (choice of): General-problem	M	sp	ind

7.5.3 *Provider-reject procedure*

This procedure is driven by the following events:

a) an unacceptable APDU as user-data of a transfer indication primitive;

b) an RORJ APDU with the Problem parameter choice General-problem as user-data of a transfer indication primitive;

c) unsuccessful APDU transfer (e.g. association abort).

7.5.3.1 *Unacceptable APDU*

The receiving ROPM receives an APDU from its peer as user data on a transfer indication primitive. If any of the fields of the APDU (except RORJ APDU) are unacceptable to this ROPM, it forms an RORJ APDU with the Problem field choice General-problem and the Invoke-ID of the rejected APDU. The receiving ROPM issues a transfer request primitive. The user-data parameter of the transfer request primitive contains the RORJ APDU.

If the received unacceptable APDU is an RORJ APDU no new RORJ APDU is formed and transferred. In this case, or after the rejection of a locally specified number of APDUs, the application-association is released abnormally.

If the application-association is not released abnormally, the receiving ROPM waits either for a transfer indication primitive from the transfer service-provider or any other primitive from the requestor.

7.5.3.2 *RORJ APDU*

The receiving ROPM receives an RORJ APDU from its peer as user-data on a transfer indication primitive. If any of the fields of the RORJ APDU are unacceptable to this ROPM, the provider-reject procedure for an unacceptable APDU is performed.

If the RORJ APDU is acceptable to the accepting ROPM and the Problem field of the RORJ APDU indicates a General-problem, it issues an RO-REJECT-P indication primitive to the acceptor. The RO-REJECT-P indication primitive parameters (Invoke-ID and Reject-reason) are derived from the RORJ APDU.

The receiving ROPM waits either for a transfer indication primitive from the transfer service-provider or any other primitive from the acceptor.

7.5.3.3 *Unsuccessful APDU transfer*

If a sending ROPM is not able to transfer an APDU by means of the transfer request primitive (e.g. in the case of abnormal association release), the sending ROPM issues an RO-REJECT-P indication primitive to the requestor for each APDU not yet transferred.

The RO-REJECT-P indication primitive parameter Returned-parameters contains the parameters of the RO-INVOKE request, RO-RESULT request, RO-ERROR request or RO-REJECT-U request primitives.

After all Returned-parameters of the APDUs not transferred have been issued to the requestor, the application-association, if it still exists, is released abnormally.

7.5.4 *Use of the RORJ APDU fields*

The RORJ APDU fields are used as follows.

7.5.4.1 *Invoke-ID*

This is the Invoke-ID field of a rejected APDU and the Invoke-ID parameter of the RO-REJECT-P indication primitive. The type and value of this field may be NULL, if the Invoke-ID field of the rejected APDU is not detectable. In this case, the Invoke-ID parameter of the RO-REJECT-P indication primitive is omitted.

7.5.4.2 *Problem: General-problem*

This is the Problem parameter value of the RO-REJECT-P indication primitive. The values used by the provider-reject procedure are:

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

- unrecognized-APDU:
  signifies that the type of the APDU, as evidenced by its Type Identifier, is not one of the four defined by this Recommendation.
- mistyped-APDU:
  signifies that the structure of the APDU does not conform to this Recommendation.
- badly-structured-APDU:
  signifies that the structure of the APDU does not conform to the standard notation and encoding, defined in Recommendations X.208 and X.209.

8 Mapping to used services

This clause defines how an ROPM transfers APDUs by means of:

a) the RTSE services, or

b) the presentation-service.

Clause 8.1 defines the mapping on the RTSE services, and clause 8.2 defines the mapping on the presentation-service.

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

8.1 *Mapping on the RTSE services*

This clause defines how RTSE service primitives described in Recommendation X.218 are used by the ROPM. Table 7/X.229 defines the mapping of the ROSE service primitives and APDUs to the RTSE service primitives.

TABLE 7/X.229

RTSE mapping overview

ROSE service	ADPU	RTSE service
RO-INVOKE request/indication	ROIV	RT-TRANSFER request/indication/confirm
RO-RESULT request/indication	RORS	RT-TRANSFER request/indication/confirm
RO-ERROR request/indication	ROER	RT-TRANSFER request/indication/confirm
RO-REJECT-U request/indication	RORJ	RT-TRANSFER request/indication/confirm
RO-REJECT-P indication	RORJ	RT-TRANSFER request/indication/confirm
Managing the Turn	-	RT-TURN-PLEASE request/indication
	-	RT-TURN-GIVE request/indication

8.1.1 *Managing the turn*

A ROPM shall possess the turn before it can use the RT-TRANSFER service. The ROPM without the turn may issue a RT-TURN-PLEASE request primitive the priority parameter of which reflects the highest priority APDU awaiting transfer.

The ROPM which has the turn, may issue an RT-TURN-GIVE request primitive when it has no further APDUs to transfer. lt will issue an RT-TURN-GIVE request primitive in response to an RT-TURN-PLEASE indication when it has no further APDUs to transfer of priority equal to or higher than that indicated in the RT-TURN-PLEASE indication primitive. If it has APDUs of lower priority still to transfer, it may issue an RT-TURN-PLEASE request whose priority reflects the highest priority APDU remaining to be transferred.

8.1.1.1 *Use of the RT-TURN-PLEASE service*

The ROPM issues the RT-TURN-PLEASE request primitive to request the turn. It may do so only if it does not already possess the turn. The RT-TURN-PLEASE service is a non-confirmed service.

The use of the RT-TURN-PLEASE service parameters is as follows:

Priority: this reflects the highest priority APDU awaiting transfer.

8.1.1.2 *Use of the RT-TURN-GIVE service*

The ROPM issues the RT-TURN-GIVE request primitive to relinquish the turn to its peer. It may do so only if it possesses the turn. The RT-TURN-GIVE service is a non-confirmed service with no parameters.

8.1.2 *APDU transfer*

Each APDU is transferred as user-data of the RT-TRANSFER service. The ROPM only issues an RT-TRANSFER request primitive, if the ROPM possesses the turn, and if there is no outstanding RT-TRANSFER confirm primitive.

8.1.2.1 *Use of the RT-TRANSFER service*

The RT-TRANSFER service is a confirmed service.

The use of the RT-TRANSFER request primitive parameters is as follows:

APDU

The APDU to be transferred. Its maximum size is not restricted in this mapping.

Transfer-time

This is specified by a local rule of the sending ROPM. lt may be related to the priority of the APDU.

The use of the RT-TRANSFER indication primitive parameters is as follows:

APDU

The APDU transferred. Its maximum size is not restricted in this mapping.

The use of the RT-TRANSFER confirm primitive parameters is as follows:

APDU

The APDU not transferred within the Transfer-time. This parameter is only provided, if the value of the Result parameter is "APDU-not-transferred". In this case the ROPM issues a RO-REJECT-P indication primitive with the parameter Returned-parameters.

Result

The parameter value "APDU-transferred" indicates a positive confirm, the parameter value "APDU-not-transferred" indicates a negative confirm.

8.2 *Mapping on the presentation-service*

This clause defines how the presentation-service primitives described in Recommendation X.216 are used by the ROPM. Table 8/X.229 defines the mapping of the ROSE service primitives and APDUs to the presentation-service primitives.

TABLE 8/X.229

Presentation-service mapping overview

ROSE service	ADPU	Presentation service
RO-INVOKE request/indication	ROIV	P-DATA request/indication
RO-RESULT request/indication	RORS	P-DATA request/indication
RO-ERROR request/indication	ROER	P-DATA request/indication
RO-REJECT-U request/indication	RORJ	P-DATA request/indication
RO-REJECT-P request/indication	RORJ	P-DATA request/indication

8.2.1 *APDU transfer*

Each APDU is transferred as user-data of the P-DATA service.

8.2.1.1 *Use of the P-DATA service*

The P-DATA service is a non-confirmed service.

The use of the P-DATA request and P-DATA indication primitive parameters is as follows:

User Data

The APDU to be transferred. Its maximum size is not restricted in this mapping.

9 Abstract syntax definition of APDUs

The abstract syntax of each ROSE APDU is specified in this clause using the abstract syntax notation of Recommendation X.208 and is shown in Figure 1/X.229.

```
Remote-Operations-APDUs {joint-iso-ccitt remote-operations(4) apdus(1)}
DEFINITIONS ::=
BEGIN
EXPORTS rOSE, lnvokeIDType;
- - the following macros are used as defined in Figure 4/X.219
IMPORTS OPERATION, ERROR FROM Remote-Operation-Notation     {joint-iso-ccitt remote-operations(4)
                                                            notation(0)}

        APPLICATION-SERVICE-ELEMENT FROM  Remote-Operations-Notation-extension
                                          {joint-iso-ccitt remote-operations(4)
                                          notation-extension(2)};

rOSE APPLICATION-SERVICE-ELEMENT ::= {joint-iso-ccitt remote-operations(4) aseID(3)}

- - APDUs
- - Types and values of operations and errors are defined in an ROSE-user protocol specification using
- - the RO-notation. Operation values are either of object identifier type or of integer type. If integer types are
- - used they shall be distinct within an abstract syntax. Error values are either of object identifier type
- - or integer types. If integer types are used they shall be distinct within an abstract syntax. There is no object
- - identifier specified for the abstrac syntax name for ROSE. However all ASN.1 data types defined in this
- - module shall be included in the named abstract syntax defined in the ROSE-user protocol specification.

ROSEapdus       ::=     CHOICE {

                            roiv-apdu [1] IMPLICIT ROIVapdu,
                            rors-apdu [2] IMPLICIT RORSapdu,
                            roer-apdu [3] IMPLICIT ROERapdu,
                            rorj-apdu [4] IMPLICIT RORJapdu}

- -  ROSE protocol specification continued
```

FIGURE 1/X.229 (Sheet 1 of 3)

Abstract syntax specification of ROSE protocol

```
- - ROSE protocol specification continued

- - APDU types

ROIVapdu        ::=     SEQUENCE {
                                invokeID InvokeIDType,
                                linked-ID [0] IMPLICIT invokedIDType OPTIONAL,
                                operation-value OPERATION,
                                argument ANY DEFINED BY operation-value OPTIONAL}
                                -- ANY is filled by the single ASN.1 data
                                -- type following the key word ARGUMENT in the type
                                -- definition of a particular operation.

InvokeIDType    ::=     INTEGER

RORSapdu        ::=     SEQUENCE {
                                InvokeID invokeIDType,
                                SEQUENCE {operation-value OPERATION,
                                        result ANY DEFINED BY operation-value
                                        -- ANY is filled by the single ASN.1 data type
                                        -- following the key word RESULT in the type
                                        -- definition of a particular operation.

                                } OPTIONAL }

ROERapdu        ::=     SEQUENCE {
                                InvokedID InvokedIDType,
                                error-value ERROR,
                                parameter ANY DEFINED BY error-value OPTIONAL}
                                -- ANY is filled by the single ASN. 1 data type following
                                -- the key word PARAMETER in the type definition
                                -- of a particular operation.

RORJapdu        ::=     SEQUENCE {
                                InvokeID CHOICE {InvokeIDType, NULL},
                                problem CHOICE {

                                        [0] IMPLICIT GeneralProblem,
                                        [1] IMPLICIT InvokeProblem,
                                        [2] IMPLICIT ReturnResultProblem,
                                        [3] IMPLICIT ReturnErrorProblem}}

- -  ROSE protocol specification continued
```

FIGURE 1/X.229 (Sheet 2 of 3)

Abstract syntax specification of ROSE protocol

```
-- ROSE protocol specification continued

GeneralProblem      ::=     INTEGER {           -- ROSE-provider detected
                                unrecognisedAPDU(0),
                                mistypedAPDU(1),
                                badlyStructuredAPDU(2)}

InvokeProblem       ::=     INTEGER {           -- ROSE-user detected
                                duplicateInvocation(0),
                                unrecognisedOperation(1),
                                mistypedArgument(2),
                                resourceLimitation(3),
                                initiatorReleasing(4),
                                unrecognisedLinkedID(5),
                                linkedResponseUnexpected(6),
                                unexpectedChildOperation(7)}

ReturnResultProblem ::=     INTEGER {           -- ROSE-user detected
                                unrecognisedInvocation(0),
                                resultResponseUnexpected(1),
                                mistypedResult(2)}

ReturnErrorProblem  ::=     INTEGER {           -- ROSE-user detected
                                unrecognisedInvocation(0),
                                errorResponseUnexpected(1),
                                unrecognisedError(2),
                                unexpectedError(3),
                                mistypedParameter(4)}

END     -- of ROSE protocol specification
```

FIGURE 1/X.229 (Sheet 3 of 3)

Abstract syntax specification of ROSE protocol

10 Conformance

An implementation claiming conformance to this Recommendation shall comply with the requirements in clauses 10.1 through 10.3.

10.1 *Statement requirements*

An implementor shall state the following:

a) the application context for which conformance is claimed, including whether the system supports the mapping of ROSE onto RTSE, onto the presentation-service, or both.

10.2 *Static requirements*

The system shall:

a) conform to the abstract syntax definition of APDUs defined in clause 9.

10.3 *Dynamic requirements*

The system shall:

a) conform to the elements of procedure defined in clause 7,

b) conform to the mappings to the used services, for which conformance is claimed, as defined in clause 8.

ANNEX A

(to Recommendation X.229)

ROPM state tables

This Annex forms an integral Part of this Recommendation.

A.1 *General*

This Annex defines a single Remote Operations Protocol Machine (ROPM) in terms of a state table. The state table shows the interrelationship between the state of an application-association, the incoming events that occur in the protocol, the actions taken, and, finally the resultant state of the application-association.

The ROPM state table does not constitute a formal definition of the ROPM. lt is included to provide a more precise specification of the elements of procedure defined in clauses 7 and 8.

This Annex contains the following tables:

a) Table A-1/X.229 specifies the abbreviated name, source, and name/description of each incoming event. The sources are:

   1) ROSE-user (ROSE-user);

   2) peer ROPM (ROPM-peer);

   3) ROPM excluding the transfer part (ROPM);

   4) transfer part of the ROPM (ROPM-TR);

   5) either Presentation service-provider (PS-provider) and the Association Control Service Element (ACSE), or the Reliable Transfer Service Element (RTSE).

b) Table A-2/X.229 specifies the abbreviated name of each state of the ROPM.

c) Table A-3/X.229 specifies the abbreviated name of each state of the ROPM-TR.

d) Table A-4/X.229 specifies the abbreviated name, target, and name/description of each outgoing event. The targets are:

1) ROSE-user (ROSE-user);

2) peer ROPM (RPOM peer);

3) ROPM excluding the transfer part (ROPM);

4) transfer part of the ROPM (ROPM-TR); and

5) either Presentation service-provider (PS-provider) and the Association Control Service Element (ACSE), or the Reliable Transfer Service Element (RTSE).

e) Table A-5/X.229 specifies the predicates.

f) Table A-6/X.229 specifies the ROPM state table using the abbreviations of the above tables.

g) Table A-7/X.229 specifies the ROPM-TR state table using the abbreviations of the above tables, if the RTSE is included in the application context.

h) Table A-8/X.229 specifies the ROPM-TR state table using the abbreviations of the above tables, if the RTSE is not included in the application context.

A.2 *Conventions*

The intersection of an incoming event (row) and a state (column) forms a cell.

In the state table, a blank cell represents the combination of an incoming event and a state that is not defined for the ROPM. (See A.3.1.)

A non-blank cell represents an incoming event and a state that is defined for the ROPM. Such a cell contains one or more action lists. An action list may be either mandatory or conditional. If a cell contains a mandatory action list, it is the only action list in the cell.

A mandatory action list contains:

a) optionally one or more outgoing events, and

b) a resultant state.

A conditional action list contains:

a) a predicate expression comprising predicates and Boolean operators (¬ represents the Boolean NOT), and

b) a mandatory action list (this mandatory action list is used only if the predicate expression is true).

A.3 *Actions to be taken by the ROPM*

The ROPM state table defines the action to be taken by the ROPM in terms of an optional outgoing event and the resultant state of the application-association.

A.3.1 *Invalid intersections*

Blank cells indicate an invalid intersection of an incoming event and state. If such an intersection occurs, one of the following actions is taken:

a) If the incoming event comes from the ROSE-user, any action taken by the ROPM is a local matter.

b) If the incoming event is related to a received APDU, PS-provider, ACSE, or RTSE; either the ROPM issues an AA-ABreq event to the ROPM-TR, or the ROPM-TR issues an ABORTreq to either the RTSE or ACSE and an AA-ABind to the ROPM.

A.3.2 *Valid intersections*

If the intersection of the state and incoming event is valid, one of the following actions is taken:

a) If the cell contains a mandatory action list, the ROPM takes the action specified.

b) If a cell contains one or more conditional action lists, for each predicate expression that is true, the ROPM takes the actions specified. If none of the predicate expressions are true, the ROPM takes one of the actions defined in § A.3.1.

TABLE A-1/X.229

Incoming event list

Abbreviated name	Source	Name and description
AA-ESTAB	RTSE	positive RT-OPEN response primitive or positive RT-OPEN confirm primitive
	ACSE	positive A-ASSOCIATE response primitive or positive A-ASSOCIATE confirm-primitive
RO-INVreq	ROSE-user	RO-INVOKE request primitive
RO-RESreq	ROSE-user	RO-RESULT request primitive
RO-ERRreq	ROSE-user	RO-ERROR request primitive
RO-RJUreq	ROSE-user	RO-REJECT-U request primitive
ROIV	ROPM-peer	valid RO-INVOKE APDU as user data on a TRANSind event
RORS	ROPM-peer	valid RO-RESULT APDU as user data on a TRANSind event
ROER	ROPM-peer	valid RO-ERROR APDU as user data on a TRANSind event
RORJu	ROPM-peer	valid RO-REJECT APDU (user -reject) as user data on a TRANSind event
RORJp	ROPM-peer	valid RO-REJECT APDU (provider-reject with General-problem) as user data on a TRANSind event
APDUua	ROPM-peer	unacceptable APDU as user data on a TRANSind event
TRANSind	ROPM-TR	transfer indication of an APDU
TRANSreq	ROPM	transfer request for an APDU
P-DATAind	PS-provider	P-DATA indication primitive
RT-TRind	RTSE	RT-TRANSFER indication primitive
RT-TRcnf+	RTSE	positive RT-TRANSFER confirm primitive
RT-TRcnf-	RTSE	negative RT-TRANSFER confirm primitive
RT-TPind	RTSE	RT-TURN-PLEASE indication primitive
RT-TGind	RTSE	RT-TURN-GIVE indication primitive
AA-REL	RTSE	RT-CLOSE response primitive or RT-CLOSE confirm primitive
	ACSE	positive A-RELEASE response primitive or A-RELEASE confirm primitive
AA-ABreq	ROPM	abort application-association
AA-ABind	ROPM-TR	application-association aborted
ABORTind	RTSE	RT-P-ABORT indication primitive or the RT-U-ABORT indication primitive
	ACSE	A-ABORT indication primitive or A-P-ABORT indication primitive

TABLE A-2/X.229

ROPM states

Abbreviated name	Name and description
STA01	idle; unassociated
STA02	associated

TABLE A-3/X.229

ROPM-TR states

Abbreviated name	Name and description
STA10	idle; unassociated
STA20	associated, token assigned, no transfer
STA21	associated, token assigned, transfer in progress
STA22	associated, token not assigned, no transfer
STA23	associated, token not assigned, transfer required
STA100	idle; unassociated
STA200	associated

TABLE A-4/X.229

Outgoing event list

Abbreviated name	Target	Name and description
RO-INVind	ROSE-user	RO-INVOKE indication primitive
RO-RESind	ROSE-user	RO-RESULT indication primitive
RO-ERRind	ROSE-user	RO-ERROR indication primitive
RO-RJUind	ROSE-user	RO-REJECT-U indication primitive
RO-RJPind	ROSE-user	RO-REJECT-P indication primitive
ROIV	ROPM-peer	RO-INVOKE APDU as user data on a TRANSreq event
RORS	ROPM-peer	RO-RESULT APDU as user data on a TRANSreq event
ROER	ROPM-peer	RO-ERROR APDU as user data on a TRANSreq event
RORJu	ROPM-peer	RO-REJECT user-reject APDU as user-data on a TRANSreq event
RORJp	ROPM-peer	RO-REJECT provider-reject APDU as user data on a TRANSreq event
TRANSreq	ROPM-TR	transfer request for an APDU
TRANSind	ROPM	transfer indication of an APDU
P-DATAreq	PS-provider	P-DATA request primitive
RT-TRreq	RTSE	RT-TRANSFER request primitive
RT-TPreq	RTSE	RT-TURN-PLEASE request primitive
RT-TGreq	RTSE	RT-TURN-GIVE request primitive
AA-ABreq	ROPM-TR	abort application-association
AA-ABind	ROPM	application-association aborted
ABORTreq	RTSE	RT-U-ABORT request primitive
	ACSE	A-ABORT request primitive

TABLE A-5/X.229

Predicates

Code	Name and description
p1	unacceptable APDU is not RORJ APDU and number of rejects does not exceed locally specified value
p2	Token initially assigned to ROPM-TR

TABLE A-6/X.229

ROPM state table

	STA01	STA02
AA-ESTAB	STA02	
RO-INVreq		ROIV STA02
RO-RESreq		RORS STA02
RO-ERRreq		ROER STA02
RO-RJUreq		RORJu STA02
ROIV		RO-INVind STA02
RORS		RO-RESind STA02
ROER		RO-ERRind STA02
RORJu		RO-RJUind STA02
RORJp		RO-RJPind STA02
APDUua		p1: RORJp STA02 ¬ p1: AA-ABreq STA01
AA-ABind		STA01
AA-REL		STA01

TABLE A-7/X.229

ROPM-TR state table for transfer by RTSE

	STA10	STA20	STA21	STA22	STA23
AA-ESTAB	p2: STA20 ¬ p2: STA22				
TRANSreq		RT-TRreq STA21		RT-TPreq STA23	
RT-TRcnf+			STA20		
RT-TRcnf-			RO-RJPind STA20		
RT-TRind				TRANSind STA22	TRANSind STA23
RT-TPind		RT-TGreq STA22	STA21		
RT-TGind				STA20	RT-TRreq STA21
AA-ABreq		ABORTreq STA10	RO-RJPind ABORTreq STA10	ABORTreq STA10	RO-RJPind ABORTreq STA10
ABORTind		AA-ABind STA10	RO-RJPind AA-ABind STA10	AA-ABind STA10	RO-RJPind AA-ABind STA10
AA-REL		STA10	RO-RJPind STA10	STA10	RO-RJPind STA10

TABLE A-8/X.229

ROPM-TR state table for transfer by presentation service

	STA100	STA200
AA-ESTAB	STA200	
TRANSreq		P-DATAreq STA200
P-DATAind		TRANSind STA200
AA-ABreq		ABORTreq STA100
ABORTind		AA-ABind STA100
AA-REL		STA100

ANNEX B

(to Recommendation X.229)

Differences between this Recommendation and Recommendation X.410-1984

This Annex is not an integral Part of this Recommendation.

This Annex describes the technical differences between the notation and protocol for Remote Operations of this Recommendation and the corresponding notation and protocol of Recommendation X.410-1984.

B.1 *Macros*

B.1.1 *New Macros*

1) *Add*: BIND macro and UNBIND macro

B.1.2 *OPERATION Macro*

1) Value Notation

Change: From: INTEGER

To: CHOICE
{ INTEGER,
OBJECT IDENTIFIER}

2) Named Type in Result production

Change: From: mandatory
To: optional

3) *Add:* Productions for linked Operations

B.1.3 *ERROR Macro*

1) Value Notation see B.1.2 item 1.

B.2 *Application protocol data units*

B.2.1 *APDUs*

1) Choice alternative

Change: From: explicit tagging
To: implicit tagging

B.2.2 *Invoke*

1) *Add:* optional linked-ID element to SEQUENCE

2) argument element

Change: From: mandatory
To: optional

B.2.3 *Return result*

1) *Add:* Field operation-value and SEQUENCE

2) result element

Change: From: mandatory
To: optional

B.2.4 *Reject*

1) Invoke Problem

Add: values (3) to (7) inclusive

B.3 *Procedures and mapping*

B.3.1 *Mapping onto used services*

1) *Add:* Mapping onto Presentation service if RTSE is absent in the Application Context

2) *Add:* Mapping for BIND and UNBIND

B.4 *Interworking between 84 and 88 implementation*

Due to B.2.1 item 1 and B.2.3 item 1 interworking between 84 and 88 implementations is not possible. However the first change was indicated in the X.400-Series Implementators Guide Version 5.

ANNEX C

(to Recommendation X.229)

Summary of assigned object identifier values

This Annex is not an integral Part of this Recommendation.

This Annex summarizes the object identifier values assigned in Recommendations X.219 and X.229.

{ **joint-iso-ccitt remote-operations(4) notation (0)** }

- - ASN.1 module defined in X.219

{ **joint-iso-ccitt remote-operations(4) apdus (1)** }

- - ASN.1 module defined in X.229

{ **joint-iso-ccitt remote-operations(4) notation-extensions (2)** }

- - ASN.1 module defined in X.219

{ **joint-iso-ccitt remote-operations(4) aseiD (3)** }

- - ASE identifier defined in X.229

{ **joint-iso-ccitt remote-operations(4) aseiD-ACSE (4)** }

- - ASE identifier defined in X.219