ETSI

EUROPEAN
TELECOMMUNICATION
STANDARD

ETS 300 134

December 1992

Source: ETSI TC-SPS

Reference: T/S 43-05

ICS: 33.080

Key words: ISDN, CCITT SS No.7, TCAP

Integrated Services Digital Network (ISDN);
CCITT Signalling System No.7
Transaction Capabilities Application Part (TCAP)

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 92 94 42 00 - Fax: +33 93 65 47 16

Contents

Foreword

This European Telecommunication Standard (ETS) has been produced by the Signalling Protocols & Switching (SPS) Technical Committee of the European Telecommunications Standards Institute (ETSI).

This ETS details exceptions and clarifications to CCITT Recommendations Q.771-Q.775 [1] - [5] defining the Transaction Capabilities (TC) of the Transaction Capabilities Application Part (TCAP) for inter-network dialogues on services such as the pan European Cellular Digital Radio System and Integrated Services Digital Network (ISDN).

Blank page

1 Scope

This ETS defines the Transaction Capabilities (TC) signalling requirements in and between networks, for non circuit related services which use the CCITT Signalling System No.7, for inter-network dialogues. Only those parts of TC need to be provided which are used by the above services.

This standard is based on CCITT Recommendations Q.771 to Q.775, normative references [1] to [5], as specified in the 1988 Blue Book. The requirements of these recommendations shall apply unless modified by the exception statements and clarifications contained in this standard.

For historical reasons, the terms TC and TCAP are used interchangeably.

2 Normative references

This ETS incorporates by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications listed hereafter. For dated references subsequent amendments to, or revisions of, any of these publications apply to this ETS only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

[1]   CCITT Recommendation Q.771 (1988): "Specifications of Signalling System No.7; Functional description of transaction capabilities".

[2]   CCITT Recommendation Q.772 (1988): "Specifications of Signalling System No.7; Transaction capabilities information element definitions".

[3]   CCITT Recommendation Q.773 (1988): "Specifications of Signalling System No.7; Transaction capabilities formats and encoding".

[4]   CCITT Recommendation Q.774 (1988): "Specifications of Signalling System No.7; Transaction capabilities procedures".

[5]   CCITT Recommendation Q.775 (1988): "Specifications of Signalling System No.7; Guidelines for using transaction capabilities".

[6]   CCITT Recommendation Q.711 (1988): "Specifications of Signalling System No.7; Functional description of the signalling connection control part".

[7]   CCITT Recommendation X.229 (1988): "Open Systems Interconnection (OSI); Remote operations: Protocol specification".

[8]   CCITT Recommendation X.208 (1988): "Open Systems Interconnection (OSI); Model and Notation: Service definition: Specification of Abstract Syntax Notation One (ASN.1)".

3 General exceptions and clarifications to CCITT Recommendations Q.771 to Q.775

3.1 Support of TC by terminal equipment

The support of TC by terminal equipment is outside the scope of this ETS.

3.2 Services assumed from a connectionless network layer

There is no requirement for TC to work over any network layer other than CCITT Signalling System No.7 Message Transfer Part (MTP) or Signalling Connection Control Part (SCCP).

The services assumed from the SCCP are provided via the N-UNITDATA and N-NOTICE primitives. Specific details of how TCAP processes the N-NOTICE indication are given in the following sections.

It is also assumed that the N-UNITDATA indication primitive shall contain the SCCP "Return Option" and "Sequence Control" parameters in addition to those parameters defined in the CCITT Recommendation Q.711 [6].

3.3   TC based on a connection-oriented network

TC based on a connection-oriented network service is outside the scope of this ETS.

3.4   Support of real-time/less real-time sensitive data

All data transfer by TC shall be considered as real-time sensitive.

3.5   Management of address information by the Transaction sub-layer

The procedures relating to address information (e.g. CCITT Recommendation Q.774 [4], subclause 3.3) are undefined in CCITT Recommendations Q.771 to Q.775 [1] to [5]. Although TC does not convey address information in any of its messages, the Transaction sub-layer must provide any necessary address information to the SCCP in every N-UNITDATA request primitive (e.g. global title and sub-system number with a "Global Title routing required" indication).

The procedures are intended to be analogous with those relating to Transaction IDs:

i)   The calling address information received in the first N-UNITDATA indication primitive in each direction of a transaction, shall be used as called address in all subsequent messages to the peer within that transaction.

ii)   Each SCCP user is responsible for providing its own address in the calling address information of every N-UNITDATA request primitive. This shall not change during the life of the transaction and shall be in a form which can be used by the SCCP to return messages, e.g. from the distant node.

iii)   Once the transaction is established, the address information shall remain constant for the life of the transaction. TCAP shall use the address information for that transaction rather than that received in subsequent N-UNITDATA indication primitives for that transaction.

NOTE 1:   In particular the above rules allow the B-SCCP-User to provide its own address as calling address information instead of the received called address information in the first N-UNITDATA indication primitive.

NOTE 2:   The encoding of calling and called party address parameter in a SCCP message must follow the rules defined in CCITT Recommendation Q.713 (1992) (see also ETS 300 009).

4   Specific exceptions and clarifications to CCITT Recommendation Q.771

NOTE: The following notation is used in the tables included in this Clause:

M   indicates a mandatory parameter;

O   indicates an optional parameter;

(=)   indicates that the parameter must have the same value in a request primitive and in the corresponding indication primitive.

4.1 Primitives for dialogue handling - Q.771, subclause 3.1.1, table 1/Q.771

In addition to the dialogue handling primitives identified in table 1/Q.771, the Component sub-layer shall support the TC-NOTICE indication primitive.

> NOTE: TC-NOTICE informs the TC-user that the service provider has been unable to provide the requested service.

4.2 Primitives for component handling - Q.771, subclause 3.1.1, table 2/Q.771

The TC-R-REJECT in table 2/Q.771 shall be used to inform the local TC-user that a component was rejected by the remote component sub-layer.

4.3 Overview of the Component sub-layer primitives - Q.771, subclause 3.1.1

The abstract syntax of parameters included in primitives must contain sufficient information to enable the concrete syntax to be encoded from the parameters supplied e.g. whether an operation/error code is local or global.

4.4 Definition of parameters within dialogue handling primitives - Q.771, subclause 3.1.2.1

The following exceptions and clarifications are made to the parameters within the Component sub-layer dialogue handling primitives.

4.4.1 Address parameters

In addition to the CCITT definition of address parameters, it shall also indicate the address type, for example a global title and sub-system number.

4.4.2 "Components Present" parameter

The reference to Q.771, subclause 3.1.3.8 should read as Q.771, subclause 3.1.3.7.

The "Components Present" parameter indicates whether or not components are present. If components are present they are delivered by TC to the TC-user in the order received from the originating TC-user.

4.4.3 "Parameters" parameter

The "Parameters" parameter is not used in any dialogue handling primitive.

4.4.4 "Quality of Service" parameter

The Quality of Service parameter shall indicate the SCCP sequence control and return option, for the SCCP connectionless network layer service as defined in CCITT Recommendation Q.711 [6].

4.4.5 "Report Cause" parameter

The "Report Cause" parameter contains information indicating the reason for the exception report, for example that the message was returned by the SCCP with the reason as specified in CCITT Recommendation Q.711 [6]. This parameter is in addition to those defined in Q.771, subclause 3.1.2.1, and is required for the TC-NOTICE indication primitive.

4.5   Dialogue facilities - Q.771, subclause 3.1.2.2

4.5.1   TC-NOTICE

In addition to the dialogue facilities mentioned in Q.771, subclause 3.1.2.2, the ability for TC-users to be notified of non-delivery of user data shall be provided by the TC-NOTICE indication primitive.

A TC-NOTICE indication primitive is only passed to the TC-user if the service requested cannot be provided (i.e. the network layer cannot deliver the message to the remote node) and the TC-user requested the return option in the "Quality of Service" parameter.

Table 1: TC-NOTICE primitive

Parameter	Primitive : TC-NOTICE
	Indication
Dialogue ID	M
Report Cause	M

4.5.2   Quality of Service

The "Quality of Service" parameter shall be optionally provided in all TC dialogue request primitives and not just the TC-BEGIN primitive as stated in Q.771.

In addition to the parameters defined by CCITT for the TC dialogue primitives (see Q.771 tables 4/Q.771, 5/Q.771, 6/Q.771, 7/Q.771 and 14/Q.771) the "Quality of Service" parameter shall be provided as indicated below, in table 2.

Table 2: "Quality of Service" requirements for TC-primitives

Quality of Service			
TC-primitive	Table	Request	Indication
TC-BEGIN	4/Q.771	O (NOTE)	M
TC-CONTINUE	5/Q.771	O (NOTE)	M
TC-END	6/Q.771	O (NOTE)	M
TC-U-ABORT	7/Q.771	O (NOTE)	M
TC-P-ABORT	14/Q.771		M
TC-NOTICE			

NOTE: When the "Quality of Service" parameter is not present in a dialogue request primitive, the Component sub-layer shall not request a quality of service to the Transaction sub-layer. The Transaction sub-layer shall request SCCP Class 0 and no return option to the SCCP.

4.6   Unstructured dialogue - Q.771, subclause 3.1.2.2.1, table 3/Q.771

The "Components Present" parameter should not be present in the TC-UNI request primitive. Table 3 shows the parameters that are required in the TC-UNI request and indication primitive.

Table 3: TC-UNI primitives

Parameter	Primitive : TC-UNI	
	Request	Indication
Quality of Service	O (NOTE)	M
Destination Address	M	M
Originating Address	M	M (=)
Dialogue ID	M	
Components Present		M

> NOTE: When the "Quality of Service" parameter is not present in a dialogue request primitive, the Component sub-layer shall not request a quality of service to the Transaction sub-layer. The Transaction sub-layer shall request SCCP class 0 and no return option to the SCCP.

4.7   End of a dialogue - Q.771, subclause 3.1.2.2.2.3.b

Components are delivered to the Component sub-layer and not directly to the Transaction sub-layer as stated in the first bullet item of Q.771, subclause 3.1.2.2.2.3.b. The text should therefore read:

"The basic scenario uses the TC-END primitives for two purposes:

-        delivery of any component(s) for which transmission is pending;

-        indication that no more components will be exchanged for this dialogue in either direction."

4.8   Report of success primitives - Q.771, subclause 3.1.3.3, table 9/Q.771

The operation code in the Return Result (Last/Not Last) components shall be included in both the request and indication primitives as shown in table 4.

Table 4: Report of success primitives

Parameter	Primitive	
	TC-RESULT-L TC-RESULT-NL Request	TC-RESULT-L TC-RESULT-NL Indication
Dialogue ID	M	M
Invoke ID	M	M (=)
Operation	O (NOTE)	O (=)
Parameters	O	O (=)
Last Component		M

NOTE: Mandatory when the primitive contains the "Parameters" parameter.

4.9   Cancel of an operation - Q.771, subclause 3.1.3.6 & figure 8/Q.771

The invocation timer is run for all classes as a local matter, but the reporting for class 4 operations is an implementation dependent matter.

4.10   Reject of a component by the Component sub-layer - Q.771, subclause 3.1.4.1

When an invalid or corrupt component is received in an END message, it is not possible to return a REJECT component and so the remote TC-user is not informed.

4.11   Primitives for transaction handling - Q.771, subclause 3.2.1, table 15/Q.771

In addition to the transaction handling primitives identified in table 15/Q.771, the Transaction sub-layer shall support the TR-NOTICE indication primitive:

NOTE: TR-NOTICE informs the TR-user that the service provider has been unable to provide the requested service.

4.12   Definition of parameters within Transaction sub-layer primitives - Q.771, subclause 3.2.1

The following exceptions and clarifications are made to the parameters within the Transaction sub-layer primitives.

4.12.1   "Quality of Service" parameter

The "Quality of Service" parameter shall indicate the SCCP sequence control and return option, for the SCCP connectionless network layer service, as defined in CCITT Recommendation Q.711 [6].

4.12.2   "Reason" parameter

The "Reason" parameter is not used in any transaction primitives.

4.12.3 "Report Cause" parameter

The "Report Cause" parameter contains information indicating the reason for the exception report, i.e. that the message was returned by the SCCP with the reason as specified in CCITT Recommendation Q.711 [6]. This parameter is in addition to those defined in Q.771, subclause 3.2.1, and is required for the TR-NOTICE indication primitive.

4.13 Transaction facilities - Q.771, subclauses 3.2.3 - 3.2.5

4.13.1 TR-NOTICE

In addition to the transaction facilities mentioned in Q.771, subclause 3.2, the ability for TR-users to be notified of non-delivery of user data shall be provided by the TR-NOTICE indication primitive. TR-NOTICE indication is given only if the transaction exists, derived from the originating Transaction ID of the returned message.

A TR-NOTICE indication primitive is only passed to the TR-user if the service requested cannot be provided (i.e. the network layer cannot deliver the message to the remote node) and the TR-user requested the return option in the quality of service parameter.

Table 5: TR-NOTICE primitive

Parameter	Primitive : TR-NOTICE
	Indication
Transaction ID	M
Report Cause	M

4.13.2 "Quality of Service"

The "Quality of Service" parameter shall be optionally provided in all TR transaction request primitives and not just the TR-BEGIN primitive as stated in Q.771. When the "Quality of Service" parameter is absent, the Transaction sub-layer shall request SCCP Class 0 and no return option to the SCCP.

In addition to the parameters defined by CCITT for the TR transaction primitives (see tables 17/Q.771, 18/Q.771, 19/Q.771, 20/Q.771 and 21/Q.771) the "Quality of Service" parameter shall be provided as indicated in table 6.

Table 6: "Quality of Service" requirements for TR-primitives

Quality of Service			
TR-PRIMITIVE	Table	Request	Indication
TR-UNI	16/Q.771	O (NOTE 1)	M (NOTE 2)
TR-BEGIN	17/Q.771	O (NOTE 1)	M (NOTE 2)
TR-CONTINUE	18/Q.771	O (NOTE 1)	M (NOTE 2)
TR-END	19/Q.771	O (NOTE 1)	M (NOTE 2)
TR-U-ABORT	20/Q.771	O (NOTE 1)	M (NOTE 2)
TR-P-ABORT	21/Q.771		M (NOTE 2)
TR-NOTICE			

NOTE 1:   When the "Quality of Service" parameter is not present in a transaction request primitive, the Transaction sub-layer shall request SCCP class 0 and no return option to the SCCP.

NOTE 2:   When this information is made available by the underlying sub-layer, then it must also be passed up to the service user.

4.14   Transaction Abort by the TR-user - Q.771, subclause 3.2.5.3

When the transaction is in the "Initiation Sent" state, i.e. a Begin message has been sent but no backward message for this transaction has been received, the result of the TR-U-ABORT request primitive is purely local. Any message subsequently received that is related to this transaction, shall be handled according to the actions indicated in table 6, of CCITT Recommendation Q.774 [4].

5 Specific exceptions and clarifications to CCITT Recommendation Q.772

5.1 Example reasons for generating P-ABORT causes - Q.772, subclause 2.3

Table 7: Example mapping of P-ABORT scenarios to P-ABORT cause values

Transaction sub-layer	
P-ABORT cause	Example reason
Unrecognised message type (Syntax error)	The combination of class, form and tag code does not correspond to a known tag i.e. message type is not Begin, Continue, End or Abort or Unidirectional.
Unrecognised transaction ID	Destination transaction ID unassigned.
Badly formatted Transaction Portion	Length indicator value less than 128 octets, but coded long form.
	Length indicator value does not correspond to length of message.
Incorrect Transaction Portion	Combination of originating and destination transaction ID does not conform to message type.
	Component Portion Tag present, but no components.
	Message does not contain all the mandatory information elements defined in CCITT Recommendation Q.773 [3] for the message type.
	The order of the received information elements within the message does not conform to CCITT Recommendation Q.773 [3] for the message type.
	Unrecognised tag other than message type.
Resource limitation	Congestion
	No Transaction ID can be allocated to the new transaction establishment request.

Table 7 identifies and gives a P-ABORT classification for those abnormal occurrences which would cause the P-ABORT message to be transmitted. The table is not exhaustive, but seeks to ensure that different administrations have a common understanding on the mapping between a P-ABORT cause and the abnormal occurrence which generates it.

The left hand side of the table shows the P-ABORT cause value as defined in Q.772, subclause 2.3. The right hand side shows examples of the type of abnormal occurrence which would generate such a message.

5.2 Operation code - Q.772, subclause 3.4

The operation code shall also be present in Return Result (Last/Not Last) components, if the result contains parameters.

5.3 Example reasons for generating General Problem reject components - Q.772, subclause 3.8.1

Table 8: Example mapping of General Problem reject scenarios to General Problem reject types

Component sub-layer	
General problem	Example reason
Unrecognised component	Component Type Tag not recognised as Invoke, Return Result Not Last Return Error, Reject or Return Result Last.
Mistyped component	Missing Invoke ID element.
	Operation code element expected but not present.
	Return Error Component received with no Error Code element.
	The order of the received information elements within the component does not conform to CCITT Recommendation Q.773 for that Component type.
Badly structured component	Length indicator value less than 128 octets but coded long form.

Table 8 identifies and gives a General Reject problem type classification for those abnormal occurrences which would cause a particular General Reject problem code to be transmitted.

The table is not exhaustive, but seeks to ensure that different administrations have a common understanding on the mapping between a General Problem type and the abnormal occurrence which generates it.

The left hand side of the table shows the General Problem code value as defined in Q.772, subclause 3.8.1. The right hand side shows examples of the type of abnormal occurrence which would generate such a component.

5.4 Mistyped parameters - Q.772, subclauses 3.8.2.3, 3.8.3.3 & 3.8.4.5

The definition given in CCITT Recommendation X.229 [7] for mistyped parameters shall be used, as this is more general and covers error conditions such as ENUMERATED error, value range error, size constraint error, value constraint error and presence constraint error.

5.4.1 Invoke problem - mistyped parameter

Signifies that the type of operation parameter supplied is not that which is agreed between the TC users. This code is generated only by the TC user.

5.4.2 Return Result problem - mistyped parameter

Signifies that the type of the result parameter supplied is not that agreed between the TC users. This code is generated only by the TC user.

5.4.3 Return Error problem - mistyped parameter

Signifies that the type of the error parameter supplied is not that agreed between the TC users. This code is generated only by the TC user.

6 Specific exceptions and clarifications to CCITT Recommendation Q.773

NOTE: The following notation is used in the tables included in this Clause:

M indicates a mandatory parameter;
O indicates an optional parameter.

6.1 Structure of the Transaction Portion - Q.773, subclause 5.1

Transaction Portion information elements shall be delivered in-sequence to the peer entity as indicated by the tables contained in this subclause.

6.2 Structure of the Component Portion - Q.773, subclause 6.1

Component Portion information elements shall be delivered in-sequence to the peer entity as indicated by the tables in this subclause.

6.3 Parameters Tag - Q.773, subclause 6.1, tables 16-18/Q.773

Tables 9 to 11 show that the parameters tag and length shall be included before any parameters.

Table 9: Invoke Component

Invoke Component	Mandatory indication
Component Type Tag Component Length	M
Invoke ID Tag Invoke ID Length Invoke ID	M
Linked ID Tag Linked ID Length Linked ID	O
Operation Code Tag Operation Code Length Operation Code	M
Parameters Tag (NOTE) Parameters Length Parameters	O

NOTE: The parameters information element may be any valid (Abstract Syntax Notation One) ASN.1 information element.

Table 10: Return Result (Last) and Return Result (Not Last) Components (NOTE 1)

Return Result (Last) and Return Result (Not Last) Component	Mandatory indication
Component Type Tag Component Length	M
Invoke ID Tag Invoke ID Length Invoke ID	M
Sequence Sequence Length	O (NOTE 2)
Operation Code Tag Operation Code Length Operation Code	O (NOTE 2)
Parameters Tag (NOTE 3) Parameters Length Parameters	O (NOTE 2)

NOTE 1:   Remote Operation Service Element (ROSE) has only one Application Protocol Data Unit (APDU) called Return Result. See CCITT Recommendation Q. 772, subclause 3.1.2.

NOTE 2:   Omitted when no information elements are included in the parameters.

NOTE 3:   The parameters information element may be any valid (Abstract Syntax Notation One) ASN.1 information element.

Table 11: Return Error Component

Return Error Component	Mandatory indication
Component Type Tag Component Length	M
Invoke ID Tag Invoke ID Length Invoke ID	M
Error Code Tag Error Code Length Error Code	M
Parameters Tag (NOTE) Parameters Length Parameters	O

NOTE: The parameters information element may be any valid (Abstract Syntax Notation One) ASN.1 information element.

6.4   Reject Component - Q.773, subclause 6.1, table 19/Q.773

Optional "parameters" are not permitted in the Reject Component. The Reject Component elements are as shown in table 12.

Table 12: Reject Component

Reject Component	Mandatory indication
Component Type Tag Component Length	M
Invoke ID Tag (Note 1) Invoke ID Length Invoke ID	M
Problem Code Tag Problem Code Length Problem Code	M

NOTE 1:   If the Invoke ID is not available, Universal Null (table 22/Q.773) with length = 0 should be used.

6.5   Corrections to the Abstract Syntax Notation (ASN) - Q.773 Annex A

Tables take precedence over ASN.1 notation. Where the tables contain ambiguities the ASN.1 description is to be used for resolution.

The Abstract Syntax Notation (ASN) for TC shall be as follows (changes to Q.773 Annex A have been sidelined):

TCAPMessages {ccitt recommendation q 773 moduleA(0)} DEFINITIONS ::=   --|

BEGIN
EXPORTS   OPERATION, ERROR;
-- Transaction Sub-Layer fields.

MesssageType ::= CHOICE{ unidirectional   [APPLICATION 1] IMPLICIT Unidirectional,   --|
                         begin   [APPLICATION 2] IMPLICIT Begin,
                         end   [APPLICATION 4] IMPLICIT End,
                         continue   [APPLICATION 5] IMPLICIT Continue,
                         abort   [APPLICATION 7] IMPLICIT Abort}

Unidirectional   ::= SEQUENCE {ComponentPortion}   --|
Begin   ::= SEQUENCE {OrigTransactionID, ComponentPortion OPTIONAL}
End   ::= SEQUENCE {DestTransactionID, ComponentPortion OPTIONAL}
Continue   ::= SEQUENCE {OrigTransactionID, DestTransactionID,
                                ComponentPortion OPTIONAL}
Abort ::= SEQUENCE { DestTransactionId,
                CHOICE { P-AbortCause, UserAbortInformation} OPTIONAL}   --|

-- NOTE: When the Abort Message is generated by the Transaction sub-layer,   --|
--   a P-AbortCause must be present.   --|

OrigTransactionID ::= [APPLICATION 8] IMPLICIT OCTET STRING
DestTransactionID ::= [APPLICATION 9] IMPLICIT OCTET STRING

```
P-AbortCause        ::= [APPLICATION 10] IMPLICIT INTEGER {                              --|
                                      unrecognizedMessageType(0),
                                      unrecognizedTransactionID(1),
                                      badlyFormattedTransactionPortion(2),
                                      incorrectTransactionPortion (3),
                                      resourceLimitation(4)}
UserAbortInformation  ::= [APPLICATION 11] EXTERNAL                                      --|
-- NOTE: The CCITT Blue Book definition of UserAbortInformation is                       --|
-- incorrect and incomplete.  In order to correct both of these limitations,            --|
-- the UserAbortInformation has been defined as EXTERNAL.  This is                       --|
-- consistent with the ACSE definition of association abort.  The EXTERNAL               --|
-- data type is defined in X.208.

-- COMPONENT PORTION.  The last field in the transaction  portion of the
-- TCAP  message is the Component Portion, if components are present.                     --|

ComponentPortion      ::= [APPLICATION 12] IMPLICIT SEQUENCE OF Component

-- Component Sub-Layer fields.

-- COMPONENT TYPE. Recommendation X.229 defines four Application Protocol
-- Data Units (APDUs).  TCAP adds returnResultNotLast to allow for the
-- segmentation of a result.  Note: in X.229 EXPLICIT rather than IMPLICIT
-- tagging is used.

Component      ::=CHOICE  {invoke            [1] IMPLICIT Invoke,
                          returnResultLast   [2] IMPLICIT ReturnResult,
                          returnError        [3] IMPLICIT ReturnError,                    --|
                          reject             [4] IMPLICIT Reject,
                          returnResultNotLast [7] IMPLICIT ReturnResult}

-- The Components are sequences of data elements

Invoke            ::= SEQUENCE {
                      invokeID INTEGER,
                      linked-ID [0] IMPLICIT INTEGER OPTIONAL,                            --|
                      operationCode OPERATION,                                           --|
                      parameter ANY DEFINED BY operationCode OPTIONAL}
                      -- ANY is filled by the single ASN.1 data type
                      -- following the key word PARAMETER in type                        --|
                      -- definition of a particular operation.

ReturnResult    ::= SEQUENCE {                                                           --|
                      invokeID INTEGER,
                      SEQUENCE { operationCode OPERATION,                                --|
                            parameter ANY DEFINED BY operationCode                       --|
                            -- ANY is filled by the single ASN.1 data
                            -- type following the key word RESULT in
                            -- the type definition of a particular
                            -- operation.
                          } OPTIONAL}

ReturnError     ::= SEQUENCE {
                      invokeID INTEGER,                                                  --|
                      errorCode ERROR,                                                   --|
                      parameter ANY DEFINED BY errorCode OPTIONAL}                       --|
                      -- ANY is filled by the single ASN.1 data type
                      -- following the key word PARAMETER in type
                      -- definition of a particular error.
```

```
Reject    ::= SEQUENCE {
                    invokeID  CHOICE { INTEGER, NULL},                           -- |
                    problem CHOICE{
                        [0] IMPLICIT GeneralProblem,
                        [1] IMPLICIT InvokeProblem,
                        [2] IMPLICIT ReturnResultProblem,
                        [3] IMPLICIT ReturnErrorProblem}}
```

-- OPERATIONS.
-- Operations are specified with the OPERATION MACRO. When an operation is
-- specified, the valid parameter set, results, and errors for that
-- operation are indicated. Default values and optional parameters are
-- permitted.

```
OPERATION MACRO ::=
BEGIN
TYPE NOTATION       ::= Parameter  Result Errors LinkedOperations            -- |
VALUE NOTATION      ::= value (VALUE CHOICE{
                                        localValue   INTEGER,
                                        globalValue  OBJECT IDENTIFIER})
Parameter           ::= "PARAMETER" NamedType │ empty                        -- |
Result              ::= "RESULT" ResultType │ empty
ResultType          ::= NamedType │ empty
Errors              ::= "ERRORS" "{"ErrorNames"}" │ empty
LinkedOperations    ::= "LINKED" "{"LinkedOperationNames"}" │ empty
ErrorNames          ::= ErrorList │ empty
ErrorList           ::= Error │ ErrorList ","Error
Error               ::= value (ERROR)  -- shall reference an error value
                        │ type      -- shall reference an error type if no error value
                                    -- is specified

LinkedOperationNames   ::= OperationList │ empty
OperationList          ::= Operation │ OperationList "," Operation
Operation              ::= value (OPERATION) -- shall reference an operation value
                           │ type      -- shall reference an operation type if no
                                       -- operation value is specified

NamedType       ::= identifier type │ type

END
```

-- ERRORS

-- Errors are specified with the ERROR MACRO. When an error is specified,
-- the valid parameters for that error are indicated. Default values and
-- optional parameters are permitted.

```
ERROR MACRO        ::=
BEGIN

TYPE NOTATION      ::= Parameter
VALUE NOTATION     ::= Value(VALUE CHOICE{
                                        localValue   INTEGER,
                                        globalValue  OBJECT IDENTIFIER})

Parameter          ::= "PARAMETER" NamedType │ empty

NamedType          ::= identifier type │ type

END
```

-- PROBLEMS

```
GeneralProblem        ::= INTEGER   {  unrecognizedComponent(0),
                                       mistypedComponent (1),
                                       badlyStructuredComponent (2)}
InvokeProblem         ::= INTEGER   {  duplicateInvokeID (0),
                                       unrecognizedOperation (1),
                                       mistypedParameter (2),
                                       resourceLimitation (3),
                                       initiatingRelease (4),
                                       unrecognizedLinkedID (5),
                                       linkedResponseUnexpected (6),
                                       unexpectedLinkedOperation (7)}
ReturnResultProblem   ::= INTEGER   {  unrecognizedInvokeID (0),
                                       returnResultUnexpected (1),
                                       mistypedParameter (2)}
ReturnErrorProblem    ::= INTEGER   {  unrecognizedInvokeID (0),
                                       returnErrorUnexpected (1),
                                       unrecognizedError (2),
                                       unexpectedError(3),
                                       mistypedParameter(4)}
```

END

7   Specific exceptions and clarifications to CCITT Recommendation Q.774

7.1   Delivery of components to the remote TC-user - Q.774, subclause 3.2.1.1.1

Components in a message shall be delivered to the remote TC-user in the same order in which they are received by the originating Component sub-layer from the local TC-user.

7.2   Operation classes - CANCEL - Q.774, subclause 3.2.1.1.3

The receipt of a TC-U-CANCEL Request primitive by the Component sub-layer, or the generation of a TC-L-CANCEL Indication primitive does not change the state of the dialogue/transaction state machine.

7.3   Operation classes - Invocation Time-out - Q.774, subclause 3.2.1.1.3

The time-out of a class 4 operation is considered a normal situation. The reporting of this occurrence to the TC-user is considered an implementation matter.

7.4   Dialogue control - Q.774, subclause 3.2.2.1

When the transaction is in the "Initiation Sent" state, i.e. a Begin message has been sent but no backward message for this transaction has been received, the result of the TR-U-ABORT request primitive is purely local. Any message subsequently received that is related to this transaction shall be handled according to the actions indicated, in table 6/Q.774.

7.5   Action taken on protocol errors in the Component Portion - Q.774 table 4/Q.774

The action taken on receipt of a corrupted component shall be as shown in table 13.

Table 13: Action taken on protocol errors in the Component Portion

			Local		Remote	
Component type received	Type of error	Local action	Component state machine	Local user advised	Component state machine	Remote user advised
INVOKE	syntax error or invalid linked ID (valid invoke ID)	initiate Reject	NA	Yes (NOTE 1)	return to idle	Yes
	syntax error (invalid invoke ID)	initiate Reject	NA	Yes (NOTE 1)	No action	Yes
RR (L/NL) or RE	syntax error (valid invoke ID)	initiate Reject	return to idle	Yes (NOTE 1)	NA	Yes
	syntax error (invalid invoke ID)	initiate Reject	no action	Yes (NOTE 1)	NA	Yes
RR (L/NL)	Op class 2/4	initiate Reject	return to idle	Yes (NOTE 1)	NA	Yes
RE	Op class 3/4	initiate Reject	return to idle	Yes (NOTE 1)	NA	Yes
REJECT	syntax error	initiate Local Reject	no action	Yes	NA	No
UNKNOWN	syntax error	initiate Reject	no action / NA (NOTE 2)	Yes (NOTE 1)	no action / NA (NOTE 2)	Yes

NA = Not Applicable

NOTE 1:   This is to alert the TC-user so it can issue a dialogue control primitive to send the Reject Component formulated by the Component sub-layer.

NOTE 2:   It is not possible to decide whether an ISM exists at that node or not, therefore no action can be taken.

7.6   Abnormal procedures relating to transaction control - Q.774, subclause 3.3.4

No reaction to transaction initiation or an established transaction are procedures relating to transaction control which might be abnormal. For example, the two ends of a transaction may be in different states, due to message loss. Such abnormal situations shall be covered by a local, implementation dependent mechanism which results in aborting the transaction locally.

8   Exceptions and clarifications to CCITT Recommendation Q.774 Annex A (SDLs)

8.1   Transaction sub-layer - Q.774 figure A-3/Q.774

8.1.1   Handling of the SCCP N-NOTICE indication primitive - Q.774 figure A-3/Q.774 Sheets 1/6, 3/6 and 4/6

The N-NOTICE indication primitive is mapped onto the TR-NOTICE indication primitive in the Idle, Initiation Sent and Active states. There is no state change. It is not valid to receive an N-NOTICE indication primitive in the Initiation Received state.

The SDLs, with modifications sidelined, are shown in figures 1,2 and 3.

8.1.2   Receipt of an ABORT message - Q.774 figure A-3/Q.774 Sheet 3/6

The receipt of an ABORT message will result in either a TR-U-ABORT or TR-P-ABORT indication primitive (see figure 2).

8.1.3   Abnormal situations - Q.774 figure A-3/Q.774 Sheet 6/6

The SDL has been corrected for receipt of an unknown message type and is shown, with modifications sidelined, in figure 4.

8.2   Dialogue handling - Q.774 figure A-4/Q.774

8.2.1   Handling of the TR-NOTICE indication primitive - figure A-4/Q.774 sheet 1/2

The TR-NOTICE indication primitive is mapped onto a TC-NOTICE indication primitive and passed to the Transaction Capabilities User. There is no state change and the state machine remains idle. The SDL is shown, with modifications sidelined, in figure 5.

8.2.2   Terminating state machines at the end of a dialogue - figure A-4/Q.774 sheets 1/2 & 2/2

At the end of a dialogue it is necessary to ensure that no invocation state machines are active or any components are awaiting transmission. This is achieved by sending a dialogue terminated primitive to the component co-ordinator. The SDLs are shown, with modifications sidelined, in figures 5 & 6.

8.3   Component co-ordinator - Q.774 figure A-5/Q.774

8.3.1   Receipt of a TC-U-Cancel request primitive - figure A-5/Q.774 Sheet 1/4

The action to be taken on receipt of a TC-U-Cancel request primitive has been clarified. The SDL is shown, with modifications sidelined, in figure 7.

8.3.2   Validation of invocation state machines - figure A-5/Q.774 Sheets 2/4 & 3/4

Before a primitive is passed to an invocation state machine, the invocation state machine is checked to see if it is active. If it is not active, the TC user is informed and a Reject Component built, except when the component itself is a Reject Component. In this case the TC user is notified by a local implementation dependent mechanism The SDLs are shown, with modifications sidelined, in figures 8 & 9.

8.3.3   Return Error Component - figure A-5/Q.774 Sheet 2/4

On receipt of a Return Error Component, an internal Return Error primitive is passed to the invocation state machine, not a Return Result-Last primitive. The SDL is shown, with modifications sidelined, in figure 8.

### 8.3.4   Discard of all subsequent components in a message - figure A-5/Q.774 Sheet 2/4

The action taken on receipt of a malformed component has been corrected and is shown in figure 9.

8.3.5   Receipt of a malformed Reject Component - figure A-5/Q.774 Sheet 2/4

On receipt of a malformed Reject Component, the TC user is informed and the component is discarded along with all subsequent components in that message. The component co-ordinator is returned to the idle state. No check can be made on active invocation state machines since a Reject Component only contains two information elements, namely the invoke id and the problem type. Both are required to determine if an active invocation state machine exists. Receipt of an invalid value is not considered a syntax error (see figures 8 and 9).

Figure 1: Modified figure A-3/Q.774 (Sheet 1/6)
Transaction sub-layer
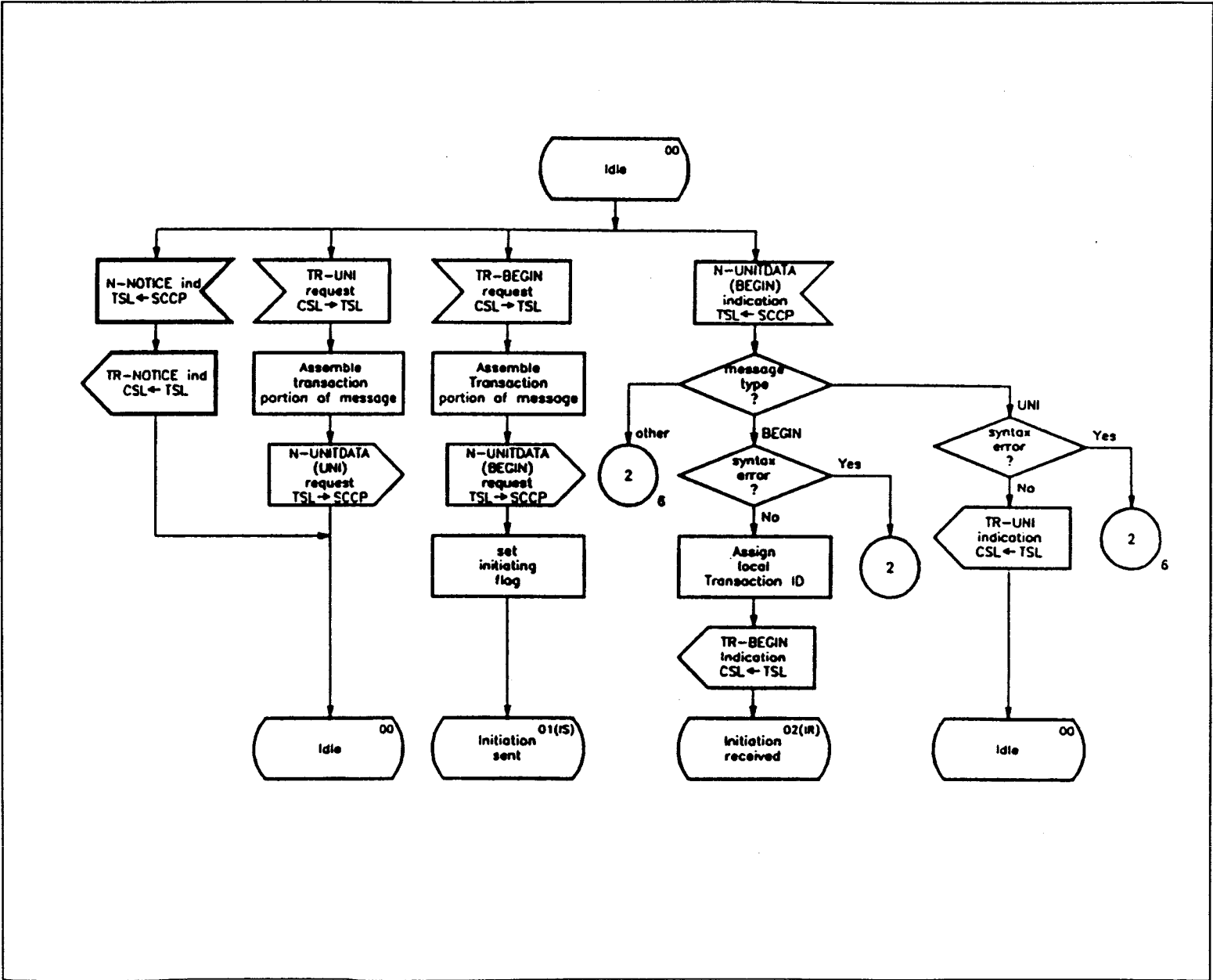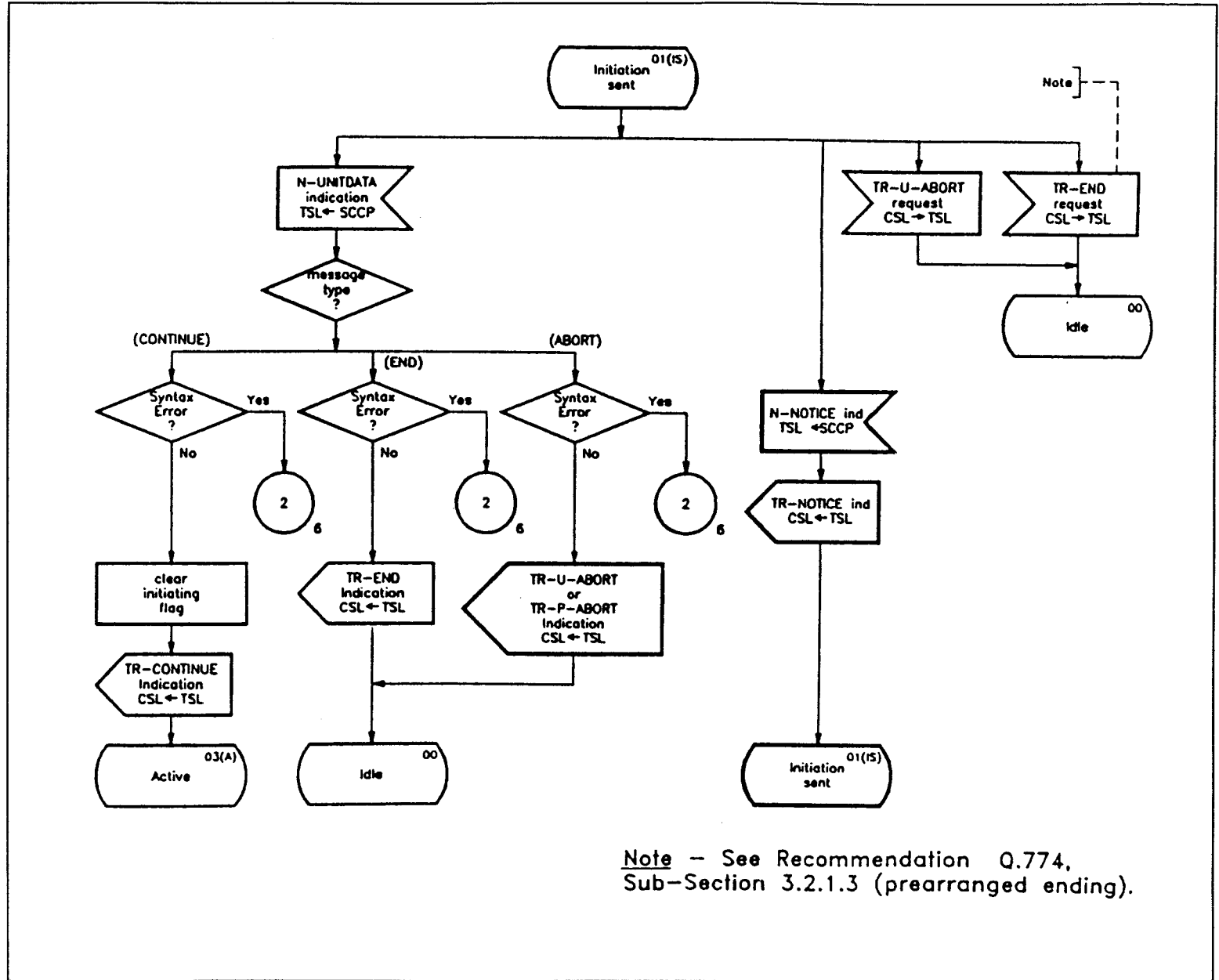
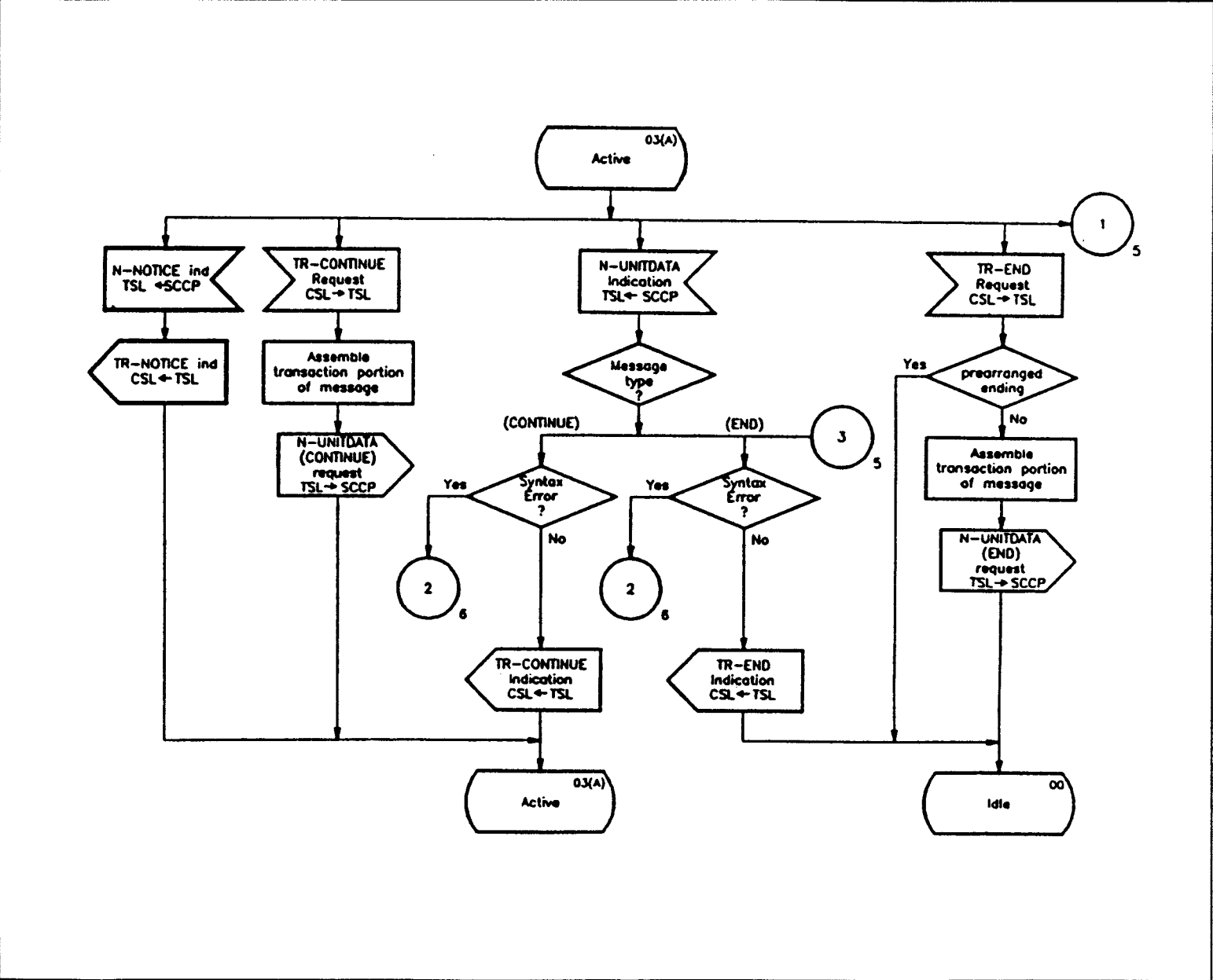Figure 2: Modified figure A-3/Q.774 (Sheet 3/6)
Transaction sub-layer

Figure 3: Modified figure A-3/Q.774 (Sheet 4/6)
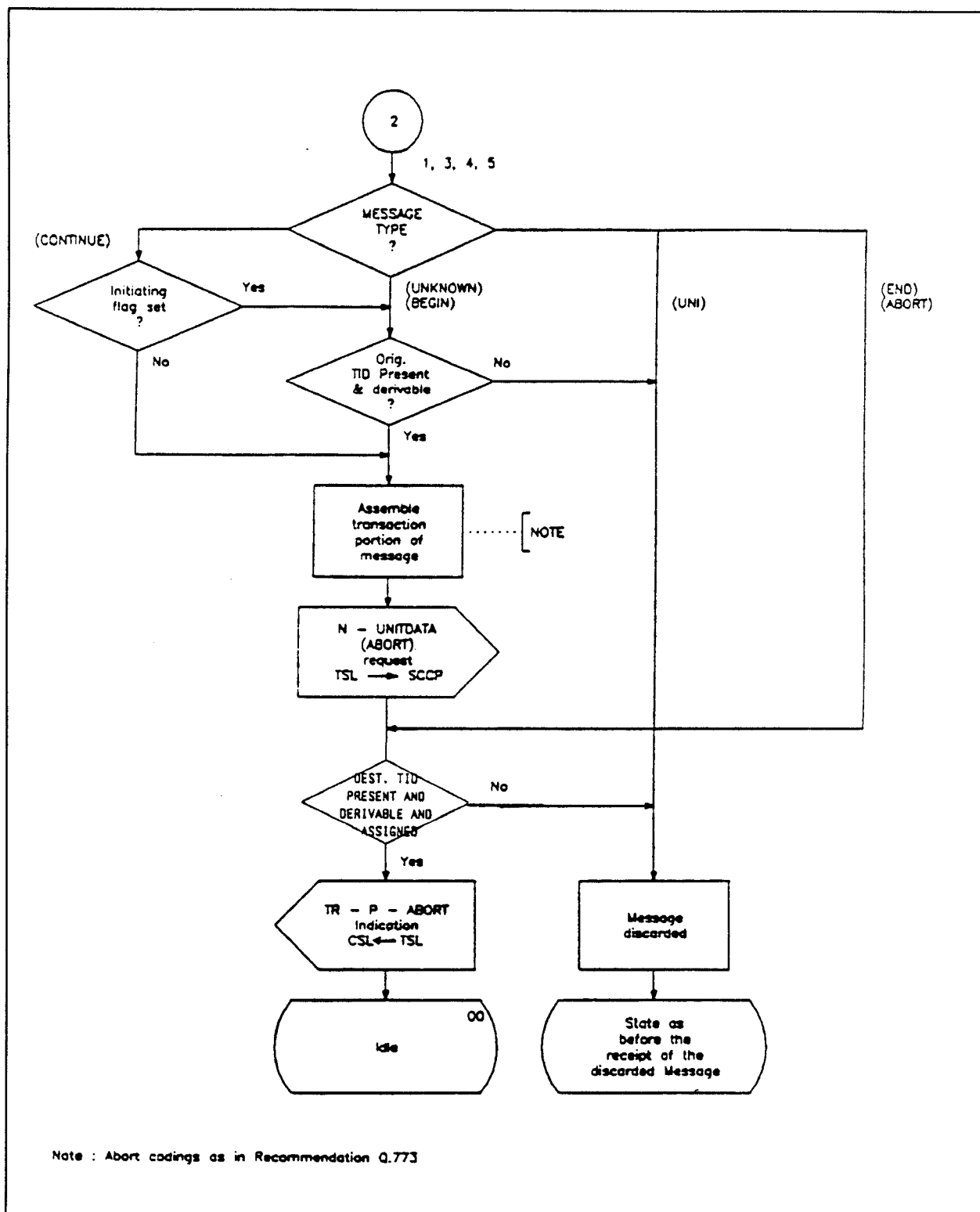Transaction sub-layer

Figure 4: Modified figure A-3/Q.774 (Sheet 6/6)
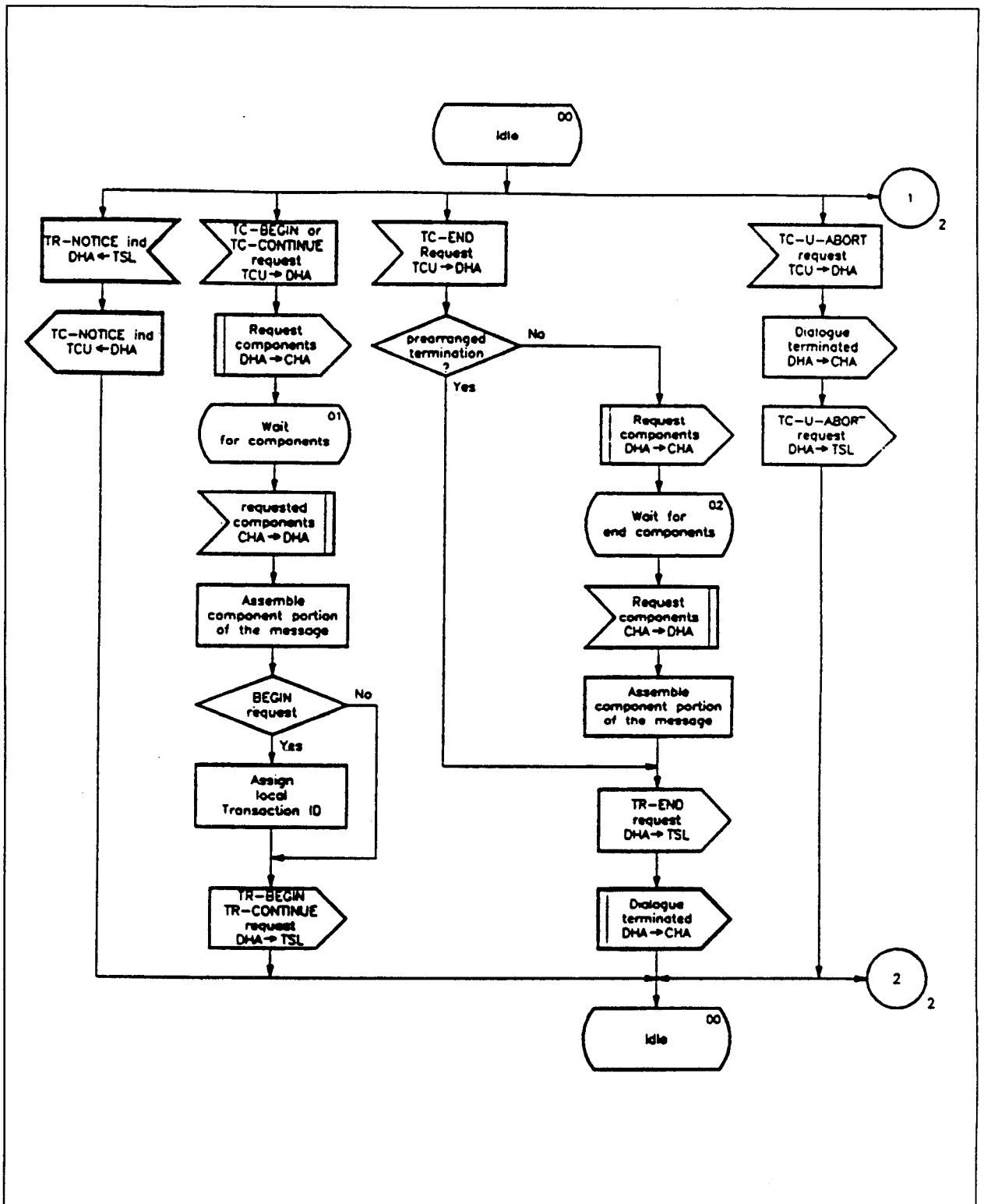Transaction sub-layer

Figure 5: Modified figure A-4/Q.774 (Sheet 1/2)
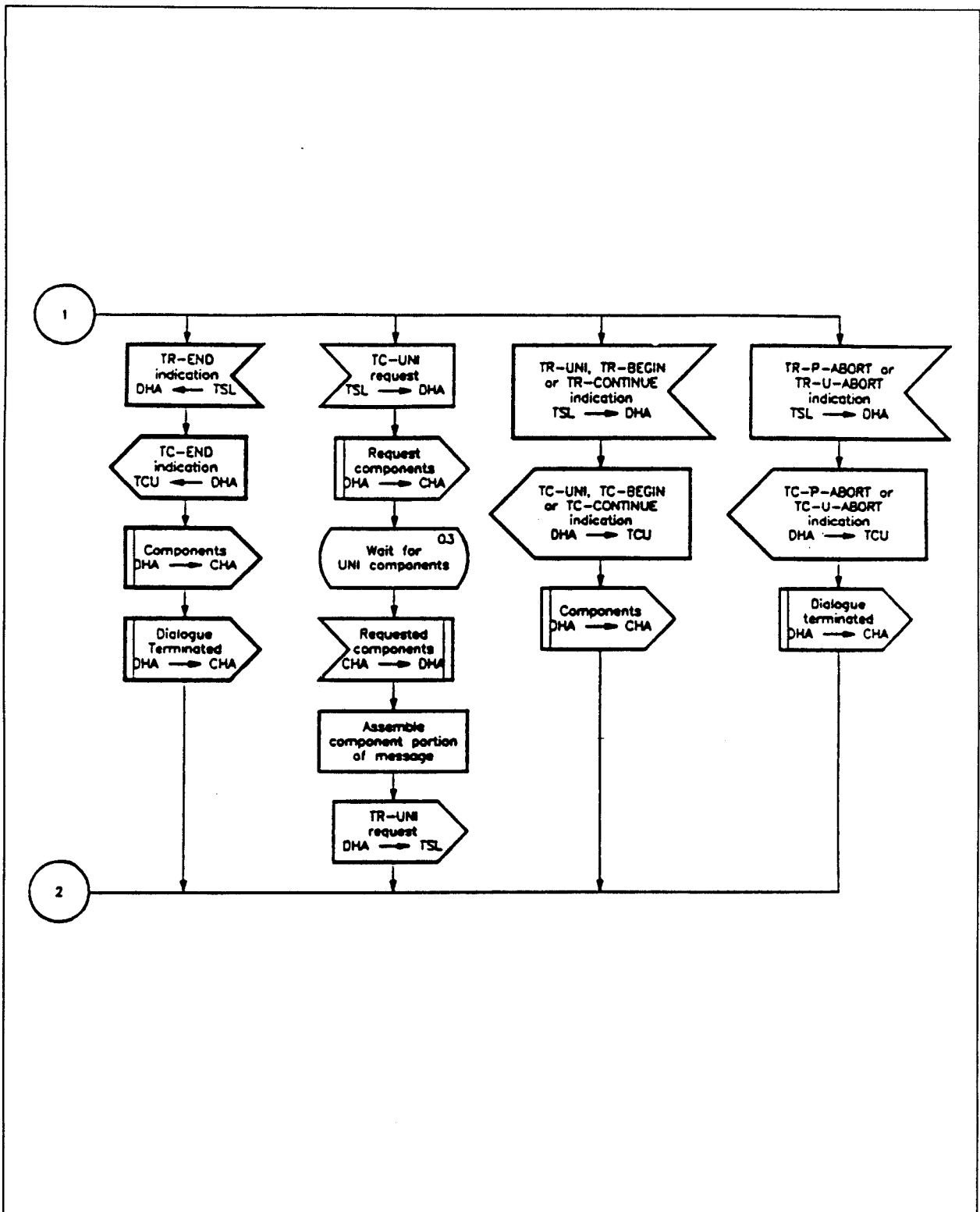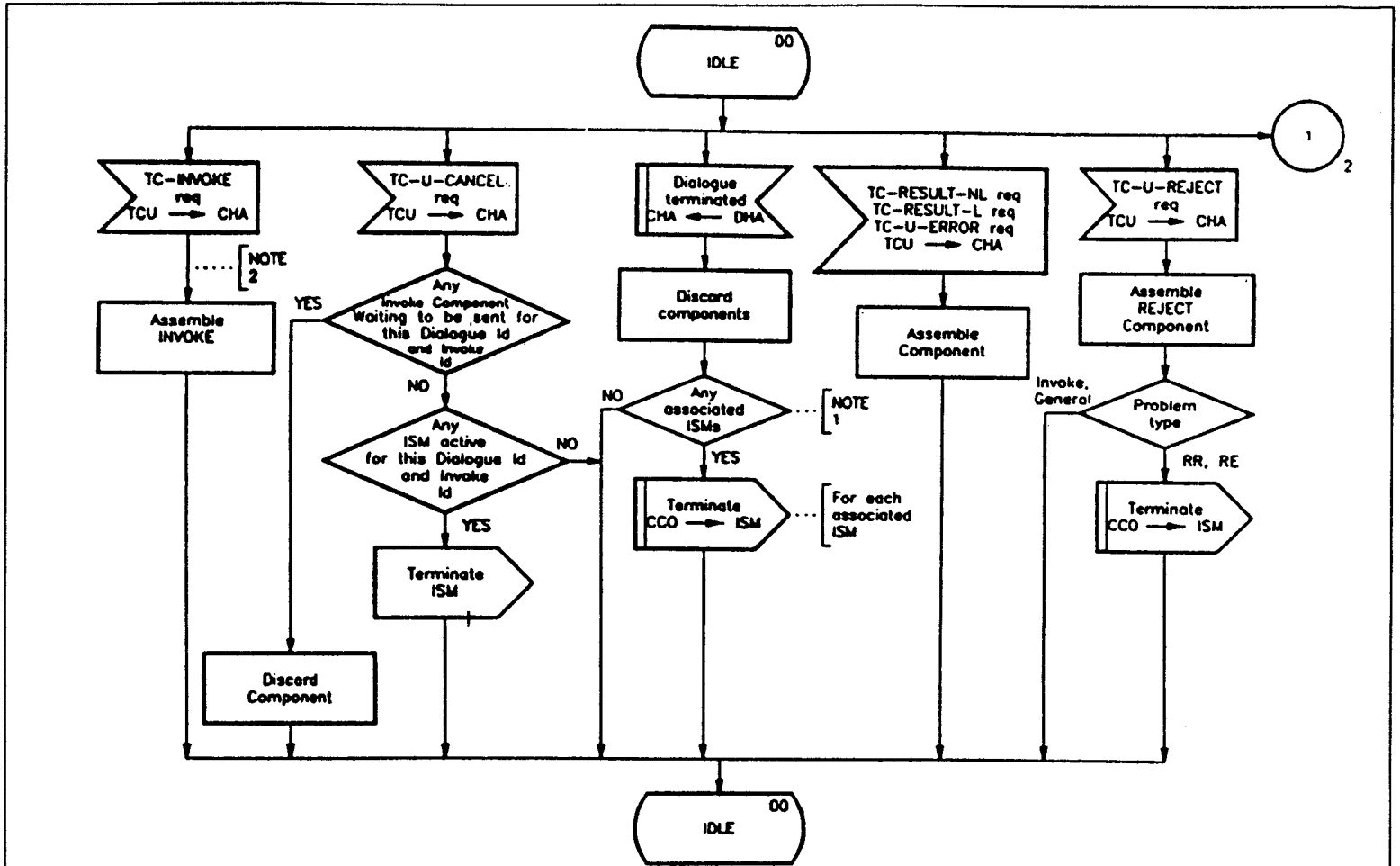Dialogue handling at the Component sub-layer

Figure 6: Modified figure A-4/Q.774 (Sheet 2/2)
Dialogue handling at the Component sub-layer

Figure 7: Modified figure A-5/Q.774 (Sheet 1/4)
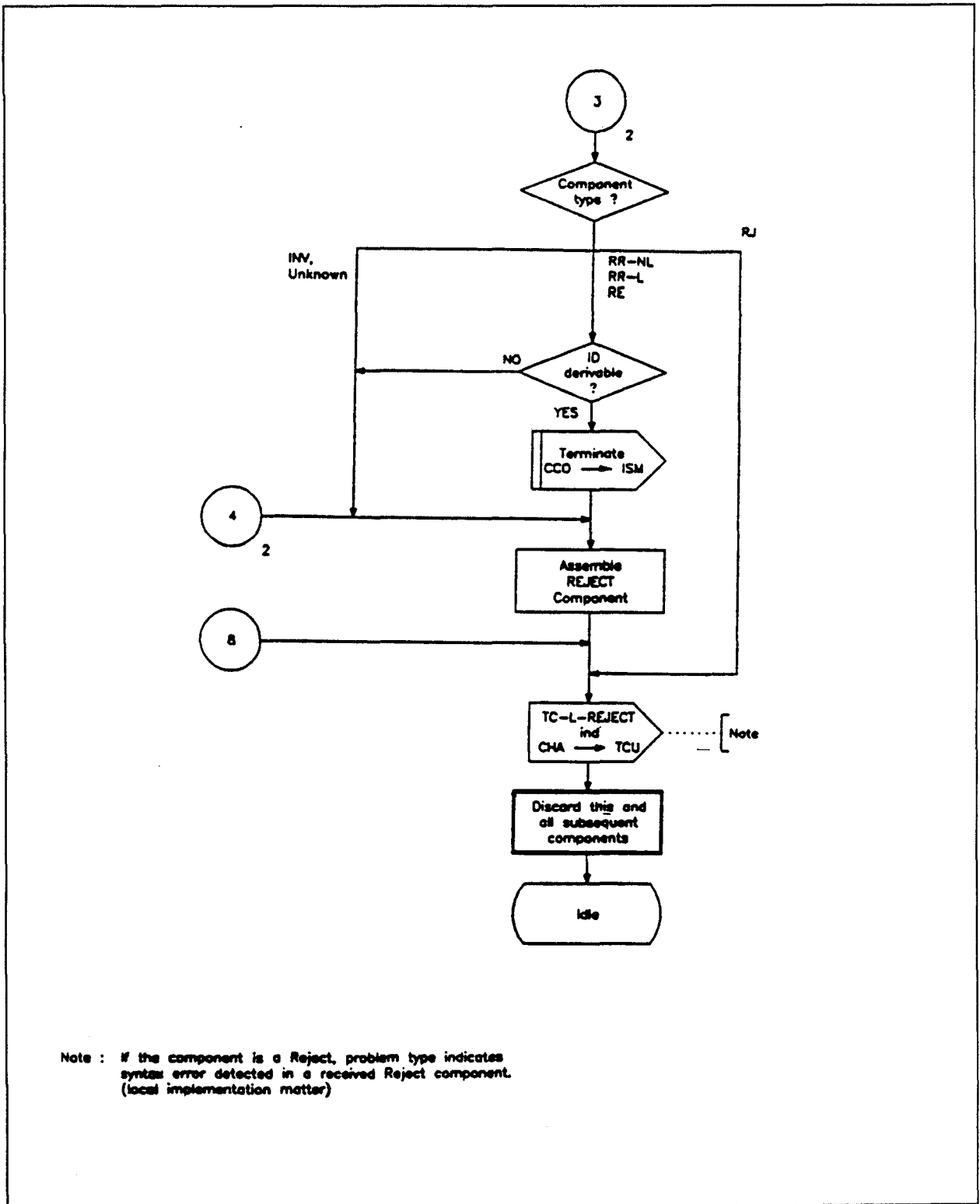Component co-ordinator

Note 1 — Implementation dependent.

Note 2 — The Invoke ID is checked to ensure associated ISM is in the Idle State and the that no other component awaiting transmission has the same Invoke ID.

Figure 8: Modified figure A-5/Q.774 (Sheet 2/4)
Component co-ordinator
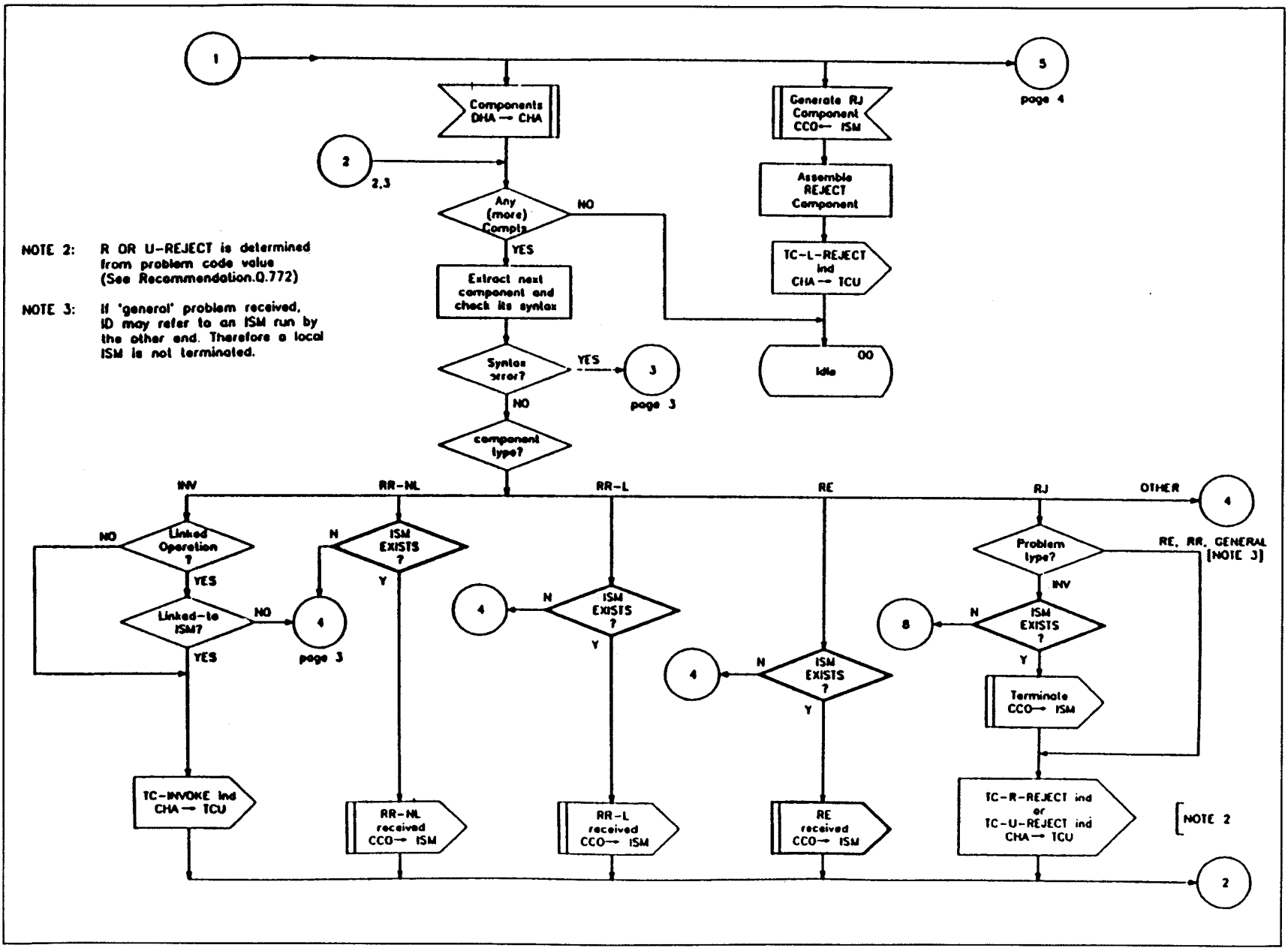
NOTE 2: R OR U-REJECT is determined from problem code value (See Recommendation.Q.772)

NOTE 3: If 'general' problem received, ID may refer to an ISM run by the other end. Therefore a local ISM is not terminated.

Figure 9: Modified figure A-5/Q.774 (Sheet 3/4)
Component co-ordinator

8.3.6   Reporting of class 4 invocation time-outs - figure A-6/Q.774 Sheet 6/6

The reporting of class 4 invocation time-outs to TC users is an implementation dependent matter and therefore may or may not result in TC-L-CANCEL indication primitive being sent to the TC-user (see figure 10).
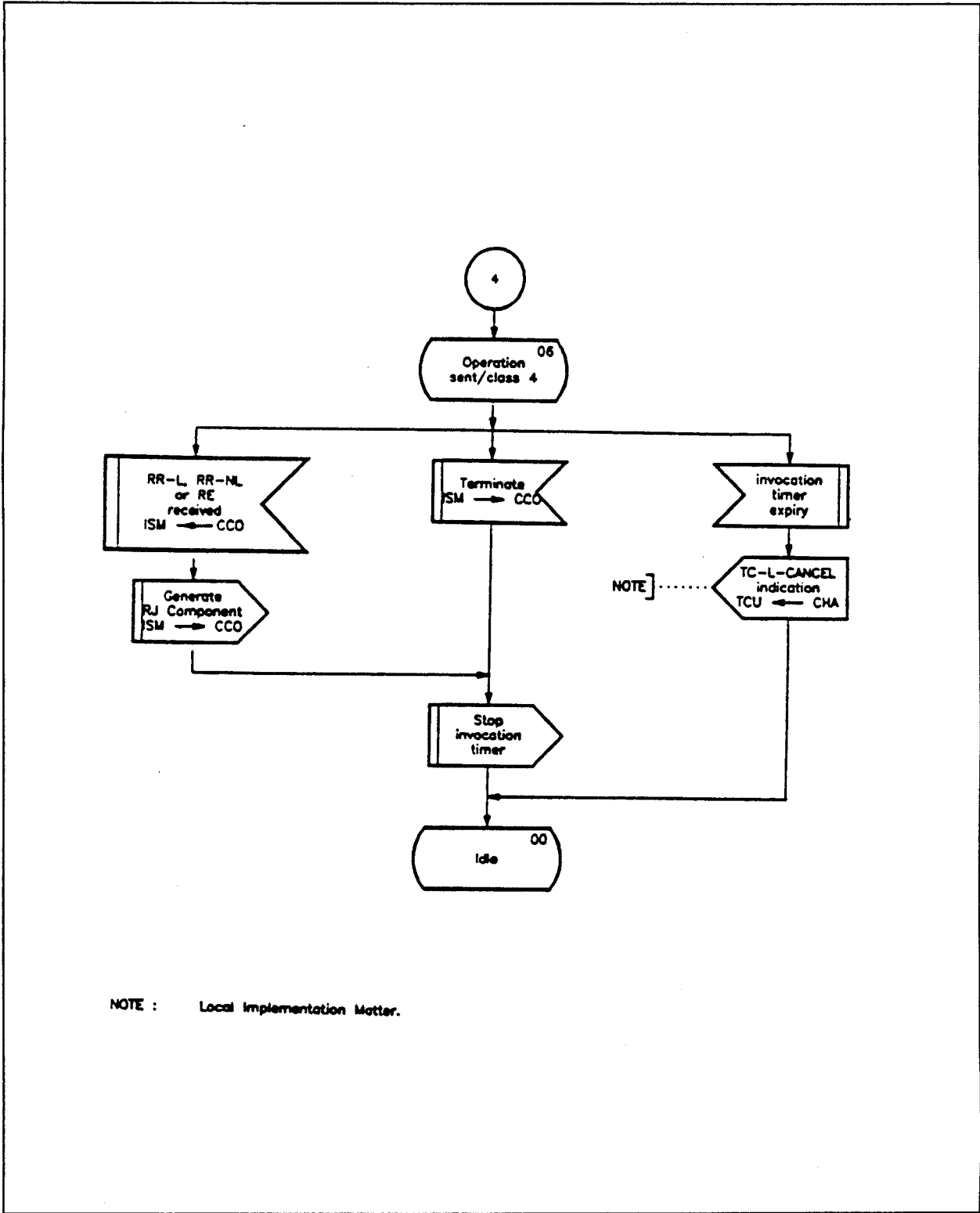
Figure 10: Modified figure A-6/Q.774 (Sheet 6/6)
Invocation State Machine

9 Specific exceptions and clarifications to CCITT Recommendation Q.775

9.1 Reject of a component by TC - Q.775, subclauses 2.4.4 & 3.2.1.3

The remote node can only be notified that a component was rejected while the dialogue is active. Therefore if it is important for an application to be notified that one of its components was rejected, then that component shall not be placed in an END message.

The dialogue is terminated by either the pre-arranged method or by sending an END message containing no components, if all the components are valid. Reject Components may be contained in either a CONTINUE or END message.

9.2 Use of the external data type for UserAbortInformation (see subclause 6.5)

Any UserAbortInformation shall be encoded EXTERNAL, as specified in CCITT Recommendation X.208 [8]. Where the encoding is a single-ASN1-type, it is recommended that at least a direct-reference or an indirect-reference is included so that the definition of the ANY data type is complete.

It should be noted that this definition of UserAbortInformation is a valid encoding which will be adopted in the 1992 version of CCITT Recommendation Q.773 (Transaction Capabilities). However, the 1988 (Blue Book) Recommendation is not consistent with this.

Annex A (informative): Symbols and abbreviations

TC	Transaction Capability
TCAP	Transaction Capability Application Part
ISDN	Integrated Services Digital Network
MTP	Message Transfer Part
SCCP	Signalling Connection Control Part
ASN.1	Abstract Syntax Notation One
ROSE	Remote Operation Service Element
APDU	Application Protocol Data Units

Annex B (informative):   Bibliography

CCITT Recommendation Q.713 (1992):   SCCP formats and codes (1992 version in preparation).

ETS 300 009:1991   Integrated Services Digital Network (ISDN); CCITT Signalling System No.7; Signalling Connection Control Part (SCCP) [connectionless service] to support international connection.

History

Document history	
December 1992	First Edition
May 1996	Converted into Adobe Acrobat Portable Document Format (PDF)