ITU Telecommunication Standardization Sector AVC-1118
Former Q.2&3/15 Rapporteur Meeting
Boston, ... 18-21 February 1997
Original: English

STUDY GROUP 16
CONTRIBUTION

Source:
M. Korpi, F.Trinkwalder, E. Horvath – Siemens
email: korpim@sbs.de
voice: +49 89 722-34570
fax: +49 89 722-23977

TITLE: Call Hold/Retrieve Supplemetary Service in H.323

Date: January 18, 1997

Purpose: To add Call Hold/Retrieve Supplementary Service to Annex of C H.323

Overview: At the last meeting of Q.2/15 in Boulder (12/96), it was agreed to specify ISO/IEC/ QSIG protocols. It was tentatively decided to start a new Annex C of H.323 for these specifications. Based on those agreements, this contribution proposes an initial base text for a specification of the call hold supplemantary service for H.323 terminals. According to the Boulder results, the text is tentatively structured in the style of an Annex C.2 of H.323; however it may be decided that some more protocol related parts of it (particularily the detailed coding) might be moved to a new Annex of H.225.0 rather than H.323.

Changes to H.323

Add to Annex C:

C.1 Call Hold

C.1.1 Scope and Field of Application

This section specifies the Call Hold supplementary service (SS-HOLD) , which is applicable to various basic services supported by H.323 Multimedia Terminals.

C.1.2 Definition

Call Hold (SS-HOLD) allows the served user, which may be the calling or the called user, to interrupt communications on an existing call and then subsequently, if desired, re-establish communications with the held party.

C.1.3 Description

SS-HOLD provides two procedures: Hold and Retrieve.

Hold may only be invoked by the served user for a call in call control state active (U10) with an existing H.245 connection. Communication on media channels is interrupted, and the call is placed in the held state. The distant party is informed, and if appropriate, a specific pattern (e.g. video/music on hold) may be provided to the held party. The served user may then originate or accept other calls or use other services without impacting the call in held state.

The served user invokes the Retrieve procedure to terminate the held state of a call. The held party is informed, and communication on media channels is re-established.

Implementations may limit the time that a call can remain in the held state. The action to be taken on timeout is outside the scope of this recommendation.

Two scenarios are specified for Hold and Retrieve:

- Near-end Call Hold

Hold is invoked at the holding endpoint as a local procedure. The holding endpoint informs the held endpoint of the hold condition by sending a ”remote holdî notification and stops receiving user packets of the held endpoint as well as sending user packets to the held endpoint. The holding endpoint may instead send music/video on hold to the held endpoint.

Retrieval is also a local procedure at the holding endpoint. The holding endpoint informs the held endpoint of the retrieval by sending a ”remote retrievalî notification.

- Remote-end Call Hold

The holding endpoint stops sending and receiving user packets and sends a hold request to the remote endpoint requiring it to stop sending and receiving user packets over that connection. The held endpoint will either accept the request and return an acknowledgement, or reject the request. Both endpoints may also inform their gatekeepers about the held connection for the purpose of bandwidth management.

Music/video on hold may be provided by the held endpoint to its local user.

A held connection may be retrieved by sending a retrieve request to the held endpoint. The held endpoint will either terminate the hold state and send an acknowledgement or reject the retrieve request.

In case of gatekeeper routed signalling, the gatekeeper forwards the requests and responses to the receiving endpoints without change. It may also use that information for its bandwidth management.

Both scenarios apply also to multipoint calls. (Details ffs)

C.1.4 Messages

For near-end call hold, a NOTIFY message shall be sent to the remote party upon hold and retrieval, as specified in H.225.0.

For remote-end call hold, FACILITY messages shall be used to convey the APDUs of the callHold and callRetrieve operations (see H.225.0).

C.1.5 Actions at the holding endpoint

C.1.5.1 Near-end Call Hold, normal procedures

The holding endpoint may invoke Hold for an existing call via a local procedure. It shall enter the held state and send a NOTIFY message containing a ”remote holdî notification to the remote endpoint.

The holding endpoint may invoke Retrieve for a call in the held state by a local procedure. It shall send a NOTIFY message containing a ”remote retrievalî notification to the remote endpoint.

The state of media connections to the remote endpoint is not changed.

C.1.5.2 Remote-end Call Hold, normal procedures

For initiating the Hold procedure, the holding endpoint shall send a FACILITY message with a holdCall invoke APDU to the remote endpoint, start timer T1 and wait for a response.

On receipt of a FACILITY message with a holdCall return result APDU timer T1 shall be stopped and the call shall be put in the held state. The bandwidth occupied by media connections of the held call may be reused for other calls.

For retrieving a held call, the holding endpoint shall send a FACILITY message with a retrieveCall invoke APDU to the held endpoint, start timer T2 and wait for a response.

On receipt of a FACILITY message with a retrieveCall return result APDU the holding endpoint shall stop timer T2, restore the bandwidth, if necessary, and terminate the held state of the call.

C.1.5.3 Exceptional procedures

If in response to a holdCall invoke APDU a FACILITY message with a holdCall return error or reject APDU is received the holding endpoint shall stop timer T1 and terminate the hold procedure. If a reject APDU is received the holding endpoint may perform the near end call hold procedure.

If timer T1 expires the holding endpoint may perform the near-end call hold procedure.

If in response to a retrieveCall invoke APDU a FACILITY message with a retrieveCall return error or reject APDU is received the holding endpoint shall stop timer T2 and release the held call.

If timer T2 expires the holding endpoint shall release the held call.

If the call in the held state is cleared by the served user or if the held endpoint terminates the H.245 connection the held state shall be terminated and normal call clearing shall apply.

C.1.6 Actions at the held endpoint

C.1.6.1 Normal procedures

On receipt of a NOTIFY message with a remote hold or remote retrieval notification the local user may be informed. Media connections associated with the held call are not affected.

On receipt of a FACILITY message with a holdCall invoke APDU, if hold is acceptable, the held endpoint shall send back a FACILITY message with a holdCall return result APDU and put the call in the held state; the bandwidth occupied by media connections of the call shall be regarded as temporarily free.

On receipt of a FACILITY message with a callRetrieve invoke APDU the endpoint shall check whether the message relates to a call in the held state. If so, the held state shall be terminated and a FACILITY message with a callRetrieve return result APDU shall be returned.

C.1.6.2 Exceptional procedures

If in the case of remote-end hold the hold request is not acceptable, the held endpoint shall send back a FACILITY message with a holdCall return error APDU.

If a retrieveCall invoke APDU received from the holding endpoint does not relate to a call in the held state, a FACILITY message wih a callRetrieve return error APDU shall be returned.

If a call in the held state is cleared by the local user or if the holding side terminates the H.245 connection for a call in the held state the held state shall be terminated and normal call clearing shall apply.

C.1.7 Dynamic Description of Near End Hold

C.1.7.1 Operational model

Hold Scenario Description from user point of view:

User A: communicating with B; put B on Hold

User B: communicating with A; being held; (music on hold, video on hold)

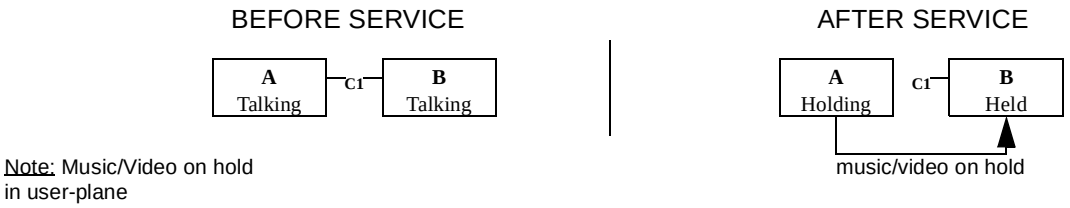


Figure C.1 -1 Operational model for near end call hold

Retrieve Scenario Description from user point of view:

User A: B on Hold; retrieve call; communicating with B

User B:being held; (music on hold, video on hold); communicating with A

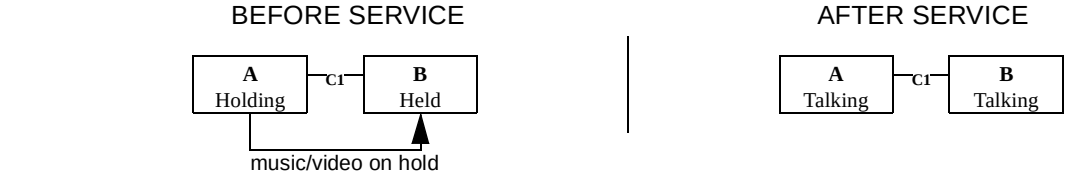


Figure C.1 -2 Operational model for near end call retrieve

C.1.7.2 Communication between Holding-NE and Holding-NE User

C.1.7.2.1 Table of primitives

generic name	type			
	request	indication	response	confirm
NOTIFY	-[1]	not defined[2]	not defined	not defined

1"-" means no parameters

2 "not defined" means that this primitive is not defined

C.1.7.2.2 Primitive definition

The NOTIFY.request primitive notifies Held-NE about setting or releasing a hold

C.1.7.2.3 Parameter Definition

No parameters.

C.1.7.2.4 States

Idle - no call hold procedure has been iniatated

NE Call held - the call is set on hold

C.1.7.2.5 State Transition Diagram

To be added

C.1.7.3 Communication between Held-NE and Held-NE User

C.1.7.3.1 Table of primitives

generic name	type			
	request	indication	response	confirm
NOTIFY	not defined[1]	-[2]	not defined	not defined

1 “not defined” means that this primitive is not defined
2“-” means no parameters

C.1.7.3.2 Primitive definition

The NOTIFY.indication primitive is used to notify the Holding-NE of the received NOTIFY.request

C.1.7.3.3 Parameter Definition

No parameters.

C.1.7.3.4 States

Idle - no call hold procedure has been iniatated

NE Call held - the call is set on hold

C.1.7.3.5 State Transition Diagram

C.1.7.4 Peer-to-peer communication for Near End Hold

C.1.7.4.1 Messages

To be added

C.1.7.4.2 Timers

To be added

C.1.7.4.3 Counters

To be added

C.1.7.4.4 Message flows

Table C.1 - 1 Near end call hold

Row no.	**User / Application action**	a) H.323 Native API primitive b) State c) Timers	Note	**IP, H.225, QSIG, CSTA, H.245**	Note	a) H.323 Native API primitive b) State c) Timers	**User / Application action**
1	**MM Terminal A**			**Network**		**MM Terminal B**	
2	Active Basic Call between TE A and TE B Capabilities exchanged H.225 connection still exists UDP path for audio open						
3	sending video, music on hold stop receiving user packets send notification toTE B	⇓ **a) Holding_NE** -<NE_Hold_req.> b)NE_Call_Held c) None		H.225 <NOTIFY> → Notification Indicator IE: QSIG: remote hold		⇓ **a) Held_NE** <NE_Hold.ind> b)NE_Call_Held c) None	Receive notification of being held stop sending user packets
4	Receive Hold Confirmation	⇓ **a) Holding_NE** -<NE_Hold_conf> b)NE_Call_Held c) None					

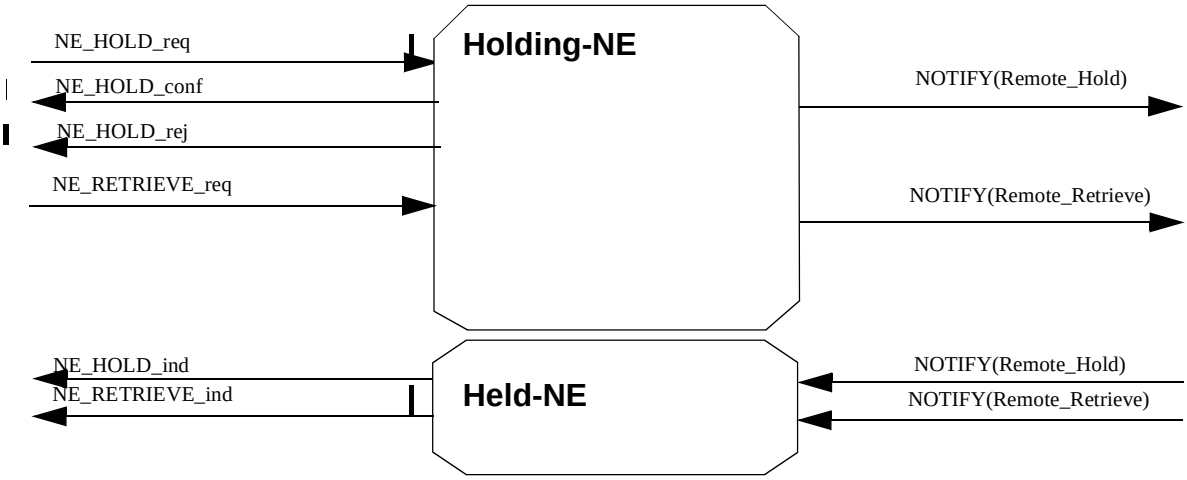
Table C.1 - 2 Near end call retrieve

Row no.	**User / Application action**	a) H.323 Native API primitive b) State c) Timers	Note	**IP, H.225, QSIG, CSTA, H.245**	Note	a) H.323 Native API primitive b) State c) Timers	**User / Application action**
5	**MM Terminal A**			**Network**		**MM Terminal B**	
6	Active Basic Call between TE A and TE B Capabilities exchanged H.225 connection still exists B on hold						

Row no.	**User / Application action**	a) H.323 Native API primitive b) State c) Timers	Note	**IP, H.225, QSIG, CSTA, H.245**	Note	a) H.323 Native API primitive b) State c) Timers	**User / Application action**
7	stop sending video, music on hold start sending / receiving user packets send notification to B	⇓ **a) Holding_NE** <NE_Retrieve.req> b)Idle c) None		H.225 <NOTIFY> ⟶ Notification Indicator IE: QSIG: remote retrieval		⇓ **a) Held_NE** -<NE_Hold.ind.> b)Idle c) None	Receive notification of being retrieved start sending user packets

C.1.7.5 SDLs

C.1.7.5.1 SDL Model Near End Hold



C.1.7.5.2 Primitive parameter default values

To be added

C.1.7.5.3 Message field default values

To be added

C.1.7.5.4 Holding-NE SDLs

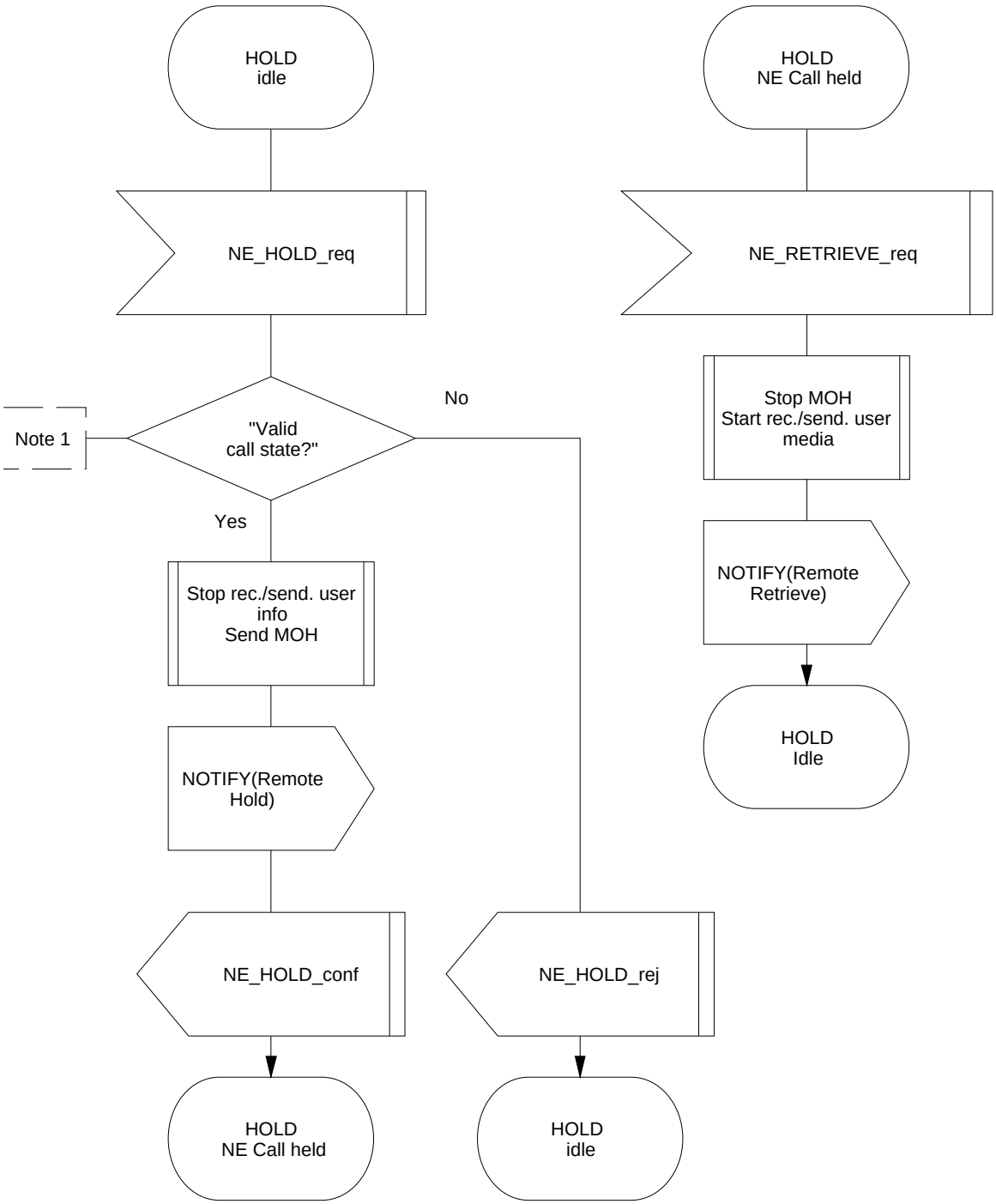


Figure C.1 -3 Holding-NE SDL

C.1.7.5.5 Held-NE SDLs

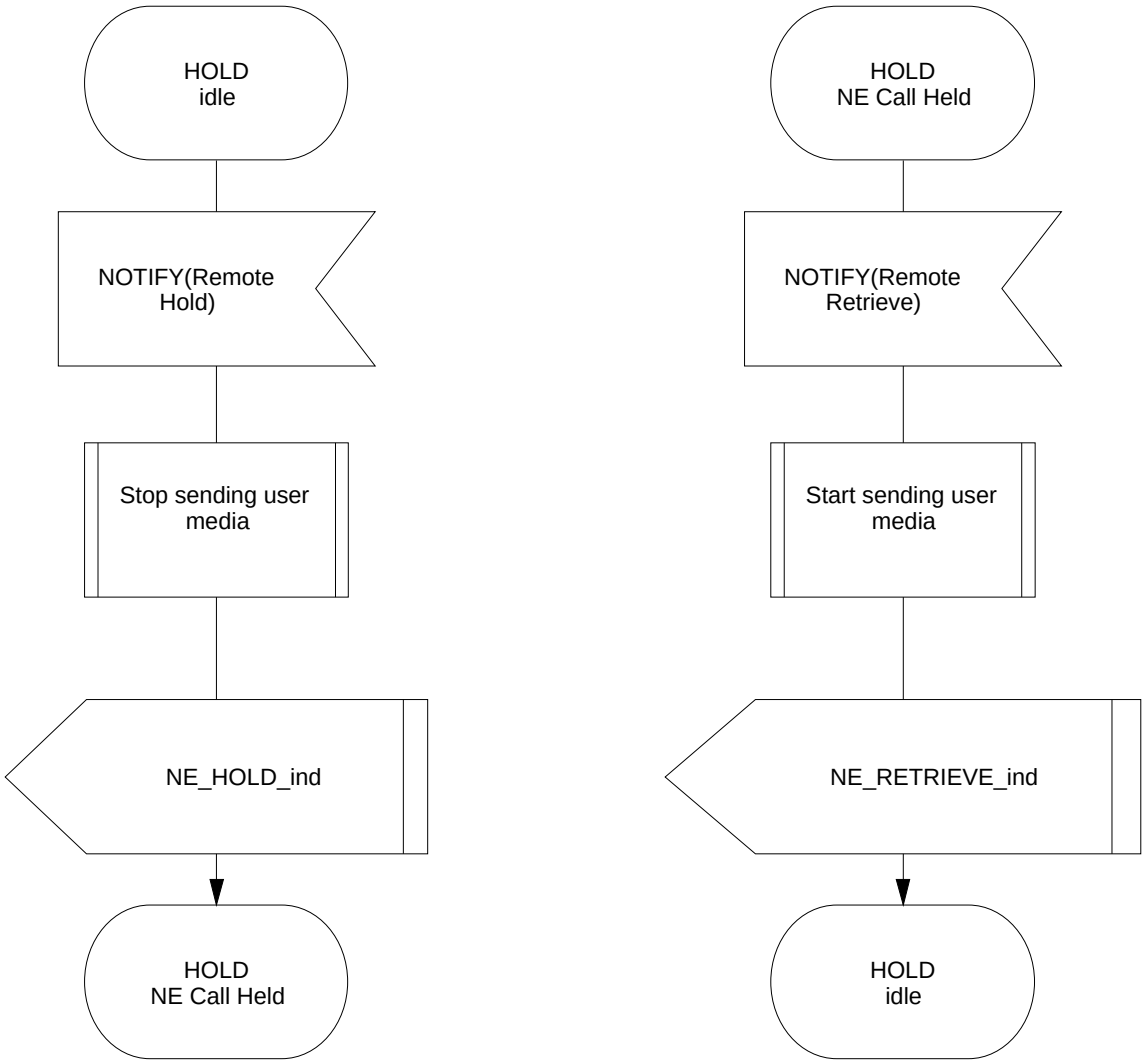


Figure C.1 -4 Held-NE SDL

C.1.8 Dynamic Description of Remore End Call Hold

C.1.8.1 Operational model

Scenario Description from user point of view**:**

User A: communicating with B; put B on Hold

User B: communicating with A; on hold (music on hold, video on hold)

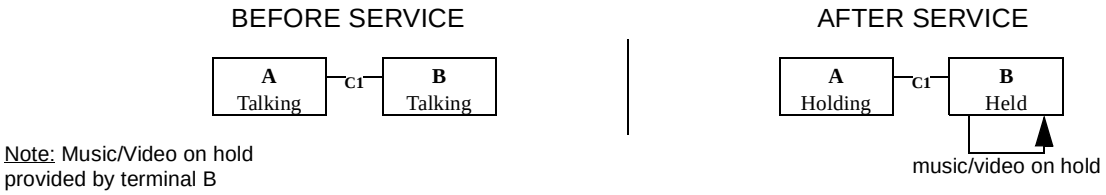


Figure C.1 -5 Operational model for remote end call hold

Scenario Description from user point of view:

User A: B on Hold; retrieve call; communicating with B

User B:being held; (music on hold, video on hold); communicating with A

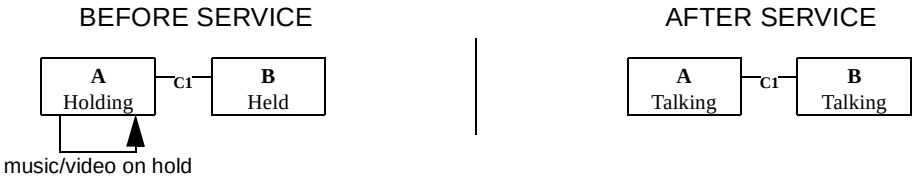


Figure C.1 -6 Operational model for remote end call retrieve

C.1.8.2 Communication between Holding-RE and Holding-RE User

C.1.8.2.1 Table of primitives

generic name	type			
	request	indication	response	confirm
HOLD	-[1]	not defined[2]	not defined	-
RERTIEVE	-	not defined	not defined	-

1 “-” means no parameters
2 “not defined” means that this primitive is not defined

C.1.8.2.2 Primitive definition

a) The HOLD.request primitive notifies Held-RE about setting or releasing a hold

b) The HOLD.confirm primitive is used to confirm the result of the hold request

c) The RETRIEVE.request primitive is used to request to retrieve the held call

d) The RETRIEVE.confirm primitive is used to confirm the result of retrieve request

C.1.8.2.3 Parameter Definition

No parameters

C.1.8.2.4 States

Idle - no call hold procedure has been iniatated

RE Hold requested - call hold request has been issued

RE Call Held - the call is set on hold

RE Retrieve requested- call retrieve request has been issued

C.1.8.2.5 State Transition Diagram

To be added

C.1.8.3 Communication between Held-RE and Held-RE User

C.1.8.3.1 Table of primitives

generic name	type			
	request	indication	response	confirm
HOLD	not defined[1]	-[2]	-	not defined
RERTIEVE	not defined	-	-	not defined

1 “not defined” means that this primitive is not defined
2 “-” means no parameters

C.1.8.3.2 Primitive definition

a) The HOLD.indication primitive is used to indicate about the hold request

b) The HOLD.response primitive is used to respond to the hold request

c) The RETRIEVE.indication primitive is used to indicate about the retrieve request

d) The RETRIEVE.response primitive is used to respond to the retrieve request

C.1.8.3.3 Parameter Definition

No parameters

C.1.8.3.4 States

Idle - no call hold procedure has been iniatated

RE Call Held - the call is set on hold

C.1.8.3.5 State Transition Diagram

To be added

C.1.8.4 Peer-to-peer communication Remote End Hold

C.1.8.4.1 Messages

To be added

C.1.8.4.2 Timers

To be added

C.1.8.4.3 Counters

To be added

C.1.8.4.4 Message Flow Model

Table C.1 - 3 Remote end call hold

Row no.	**User / Application action**	a) H.323 Native API primitive b) State c) Timers	Note	**IP, H.225, QSIG, CSTA, H.245**	Note	a) H.323 Native API primitive b) State c) Timers	**User / Application action**
7	**MM Terminal A**			**Network**		**MM Terminal B**	
8	Active Basic Call between TE A and TE B Capabilities exchanged H.225 connection still exists UDP path for audio open						
9	request to put TE B on hold	⇓ **a) Holding_RE** -<RE_Hold.req.> b)RE_Hold_Request		H.225 <FACILITY> → Facility IE: invoke holdCall		⇓ **a) Held_RE** -<RE_Hold .ind.> b)RE_Call_Held	Receive hold request
10	receive hold acknow-ledgement	⇓ **a) Holding_RE** -<RE_Hold.conf.> b)RE_Call_Held		H.225 <FACILITY> ← Facility IE: returnResult holdCall			send hold acknowledge-ment provide video, music on hold by TEB

Table C.1 - 4 Remote end call retrieve

Row no.	**User / Application action**	a) H.323 Native API primitive b) State c) Timers	Note	**IP, H.225, QSIG, CSTA, H.245**	Note	a) H.323 Native API primitive b) State c) Timers	**User / Application action**
11	**MM Terminal A**			**Network**		**MM Terminal B**	
12	Held Call between TE A and TE B Capabilities exchanged H.225 connection still exists B on hold						
13	request to retrieve B	⇓ **a) Holding_RE** <RE_retrieve.req.> b)RE_Retrieve_requested c) Start Timer Txxx		H.225 <FACILITY> → Facility IE: invoke retrieveCall		⇓ **a) Held_RE** <RE_retrieve Call.ind.> b)Idle	Receive retrieve request
14	receive retrieve acknowledgement	⇓ **a) Holding** <RE_retrieve.conf> b)Idle c) Stop Timer Txxx		H.225 <FACILITY> ← Facility IE: returnResult retrieveCall			send retrieve acknowledgement stop video, music on hold

C.1.8.5 SDLs

C.1.8.5.1 **Model** Remote End Hold

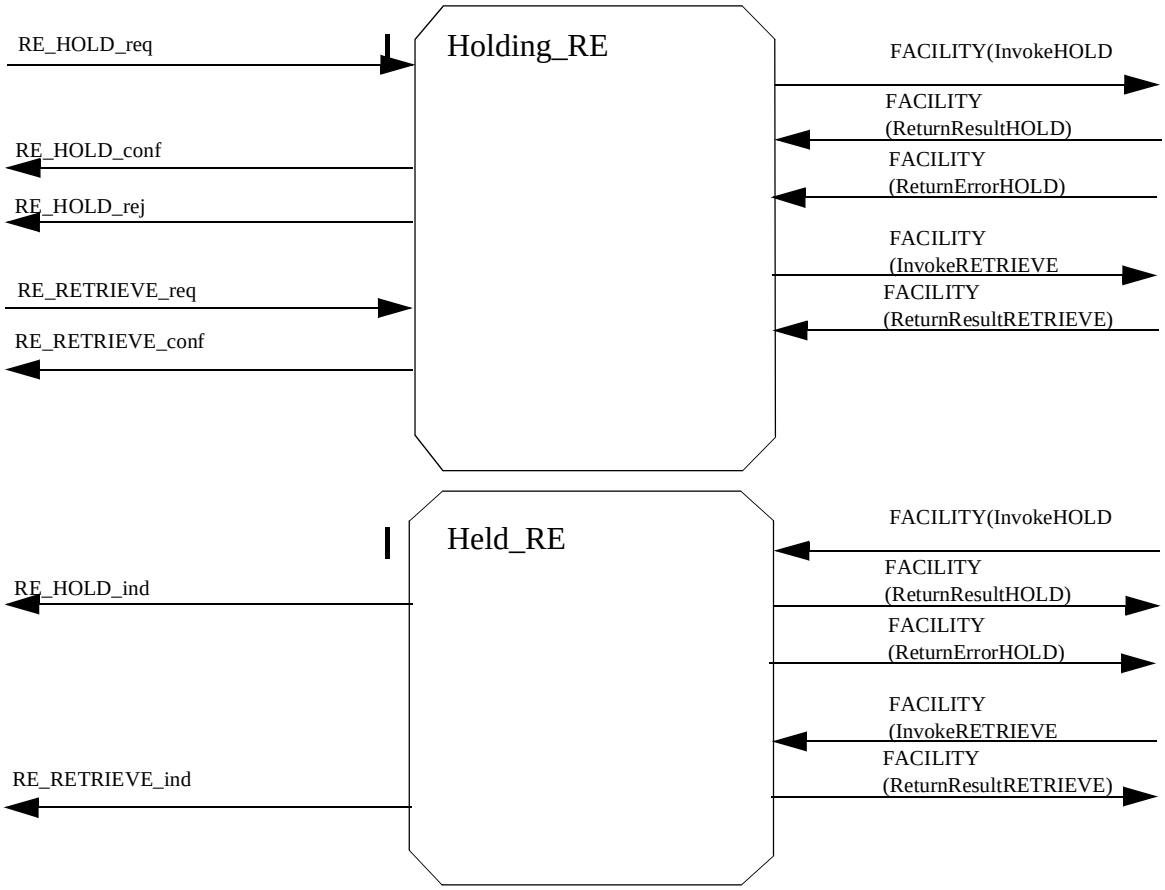


Figure C.1 -7 Model Remote End Hold

C.1.8.5.2 **Primitive parameter default values**

To be added

C.1.8.5.3 **Message field default values**

To be added

C.1.8.5.4 Holding-RE SDLs

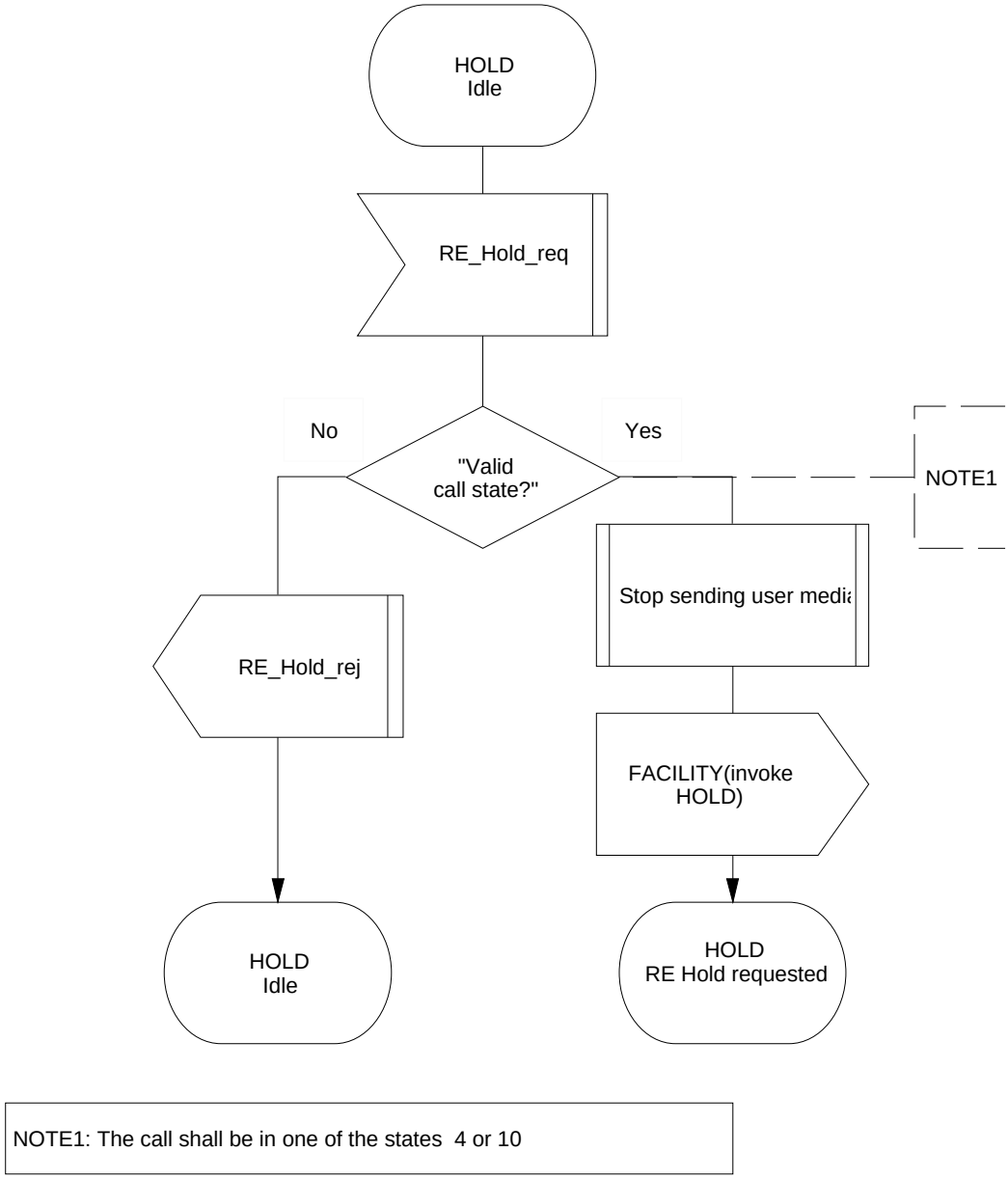


Figure C.1 -8 Holding-RE SDL (1 of 4)

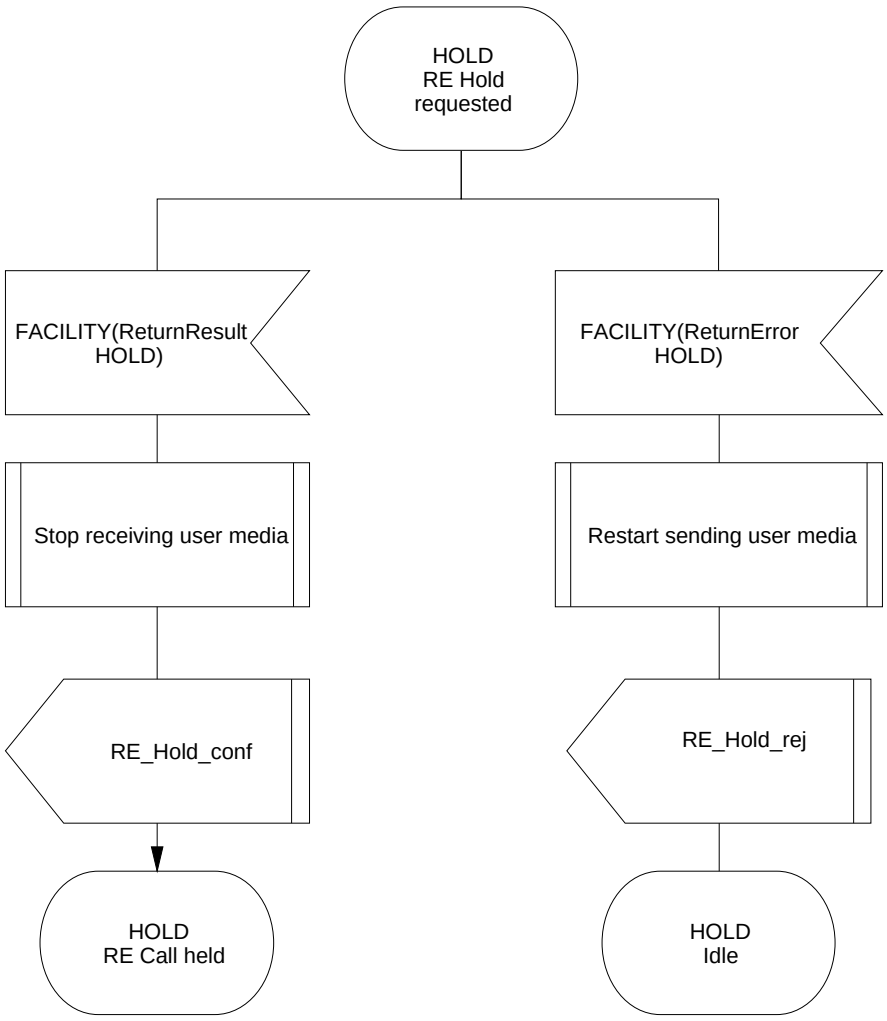


Figure C.1 -9 Holding-RE SDL (2 of 4)

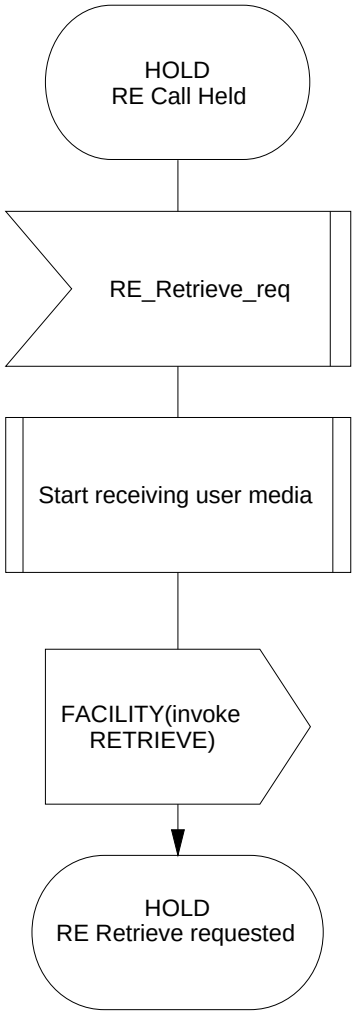


Figure C.1 -10 Holding-RE SDL (3 of 4)

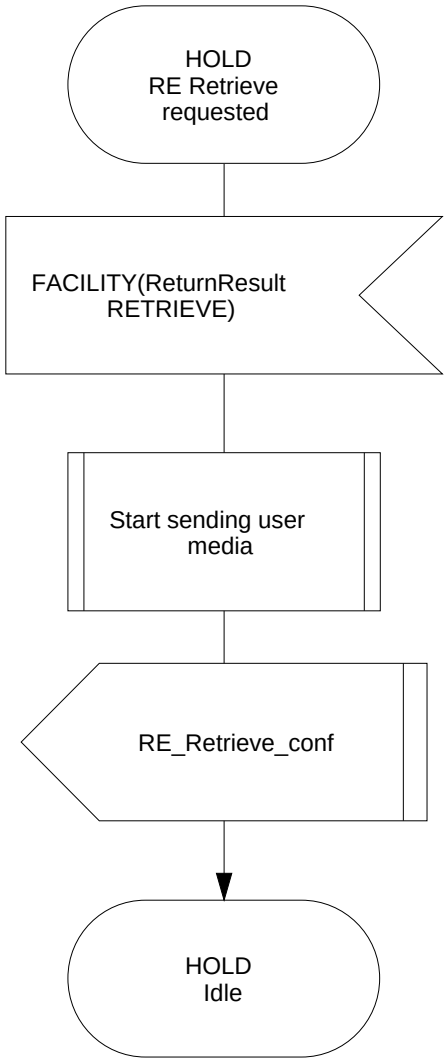


Figure C.1 -11 Holding-RE SDL (4 of 4)

C.1.8.5.5 Held-RE SDLs

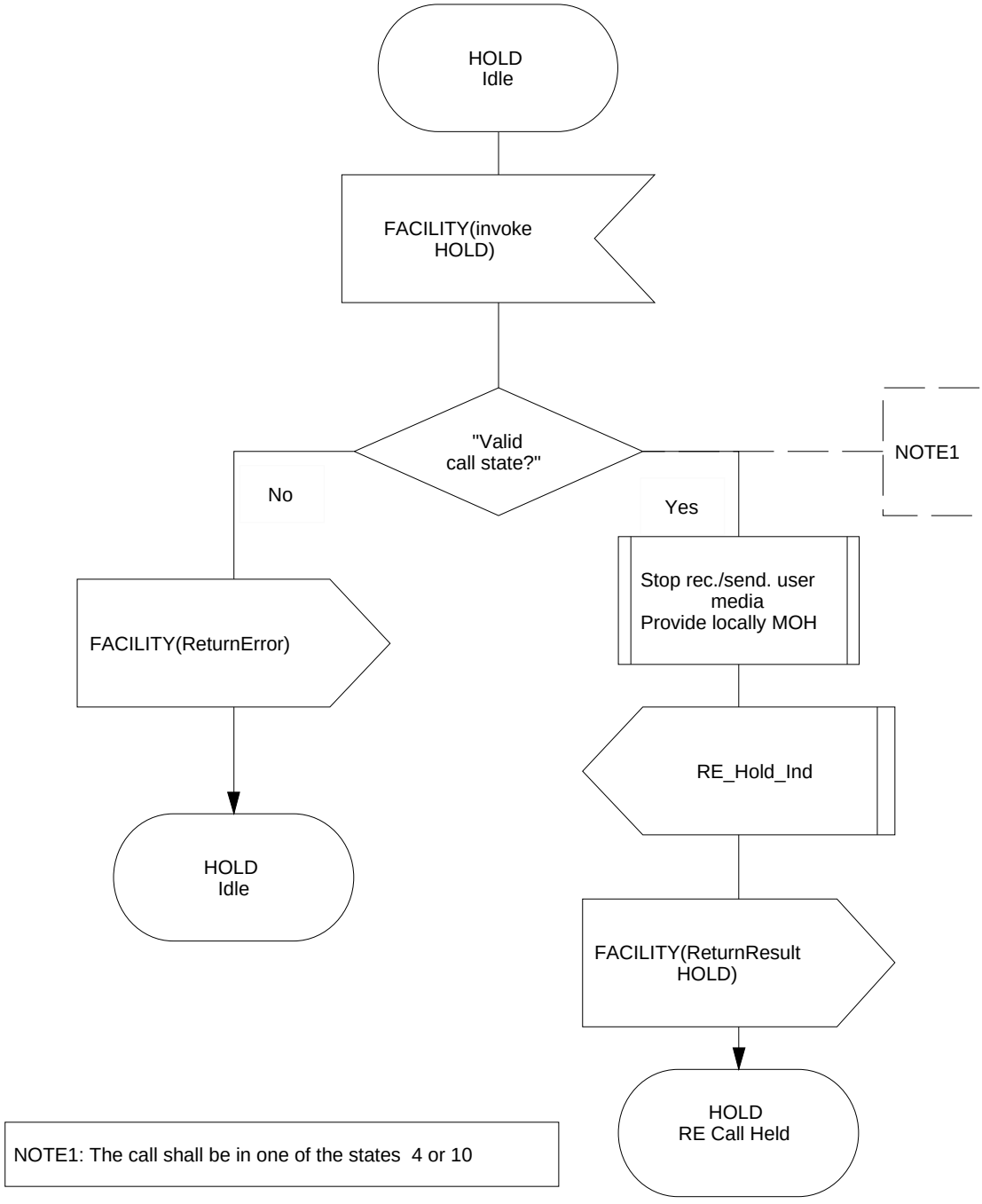


Figure C.1 -12 Held-RE SDL (1 of 2)

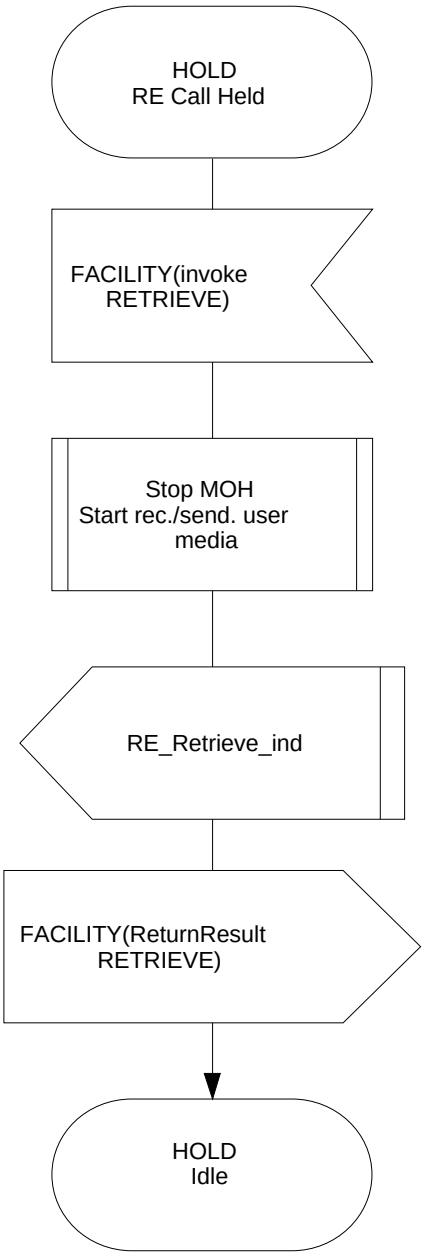


Figure C.1 -13 Held-RE SDL (2 of 2)

Changes to H.225.0

Add to Annex X:

X.1 Operations for Call Hold

```
H323-CALL-HOLD-OPERATIONS DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
    IMPORTS     OPERATION, ERROR FROM Remote-Operations-Notation
                    { joint-iso-itu-t (2) remote-operations (4) notation (0) },
                Extension FROM Manufacturer-specific-service-extension-definition
                    { iso (1) standard (0) pss1-generic-procedures (11582) msi-definition (0) },
                notAvailable, invalidCallState, resourceUnavailable, notAllowed,
                    supplementaryServiceInteractionNotAllowed FROM General-Error-List
                    { itu-t recommendation q 950 general-error-list (1) };
    HoldCall    ::=OPERATION-- requests from remote endpoint to hold the call
                    ARGUMENT
                    RESULT
                    ERRORS
                    {
                    notAvailable, -- hold not available in combination with the basic service
                    invalidCallState, -- hold not possible in current call state
                    resourceUnavailable, -- maximum number of calls on hold reached
                    notAllowed, -- hold not allowed for this call
                    supplementaryServiceInteractionNotAllowed, -- other s.s. prohibits hold
                    unspecified --implementation specific
                    }
    RetrieveCall    ::=OPERATION-- requests from remote endpoint to retrieve the call
                    ARGUMENT
                    RESULT
                    ERRORS
                    {
                    invalidCallState, -- call is not in the held state
                    resourceUnavailable, -- not enoughbandwidth
                    unspecified --implementation specific
                    }
    Unspecified     ::=ERROR
                    PARAMETER Extension
    holdCall        HoldCall::=t.b.a.
    retrieveCall    RetrieveCall::=t.b.a.
    unspecified     Unspecified::= 1008
END-- of H323-Call-Hold-Operations
```