3GPP TSG CT Plenary Meeting #28 **CP-050159**
01-03 June 2005, Quebec, CANADA

Source: **CT5 (OSA)**

Title: **2 Rel-6 CR 29.199-03**

Agenda item: **9.7 (OSA Enhancements [OSA3])**

Document for: **APPROVAL**

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Doc-2nd-Level	Workitem
CP-050159	29.199-03	0001	-	Rel-6	Incorrect method mames in WSDL files for Call Notification Interface	F	6.0.0	C5-050246	OSA3
CP-050159	29.199-03	0002	-	Rel-6	Add display name data	F	6.0.0	C5-050294	OSA3

CR-Form-v7.1

CHANGE REQUEST

⌘ **29.199-03** **CR** **0001** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Incorrect method mames in WSDL files for Call Notification Interface
Source:	⌘	CT5 Jörgen Dyst (Appium - Parlay Member)
Work item code:	⌘	OSA3 — ***Date:*** ⌘ 28/04/2005
Category:	⌘	**F** — ***Release:*** ⌘ *REL-6*

Use one of the following categories:
- ***F*** *(correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(addition of feature),*
- ***C*** *(functional modification of feature)*
- ***D*** *(editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	The semantics of the handlexxx operations in the call notification namespace defined in the WSDL are not defined in the 29.199-03 6.00. On the other hand the notifyxxx operations in 29.199-03 clause 8.2 are not represented at all in the WSDL, which makes the WSDL for the call notification API inconsistent with the operation definitions in the specification. The WSDL files use incorrect operation names for Call Notification API
Summary of change:	⌘	Change all occurrencies of operation names “handlexxx” into “notifyxxx” in the WSDL Files: parlayx_call_notification_service_2_0.wsdl and parlayx_call_notification_interface_2_0.wsdl in accordence with TS 29.199-3 clause 8.2 “Interface: CallNotification”. (xxx represents: Busy, NotReachable, NoAnswer, CalledNumber)
Consequences if not approved:	⌘	The notifyxxx operations in 29.199-03 are not represented at all in the WSDL, which makes it impossible to implement the call nofication API

Clauses affected:	⌘	Annex A (reference to WSDL files) The numbering of the attachment file names will have to be changed wsdl files: parlayx_call_notification_service_2_0.wsdl and parlayx_call_notification_interface_2_0.wsdl

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

Annex A (normative):
WSDL for call notification

The document/literal WSDL representation of this interface specification is compliant to 3GPP TS 29.199-1 [6] and is contained in text files (contained in archive 29199-03-6100-doclit.zip) which accompanies the present document.

Annex B (informative):
Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Dec 2003	CN_21	NP-030552	--	--	Submitted to CN#22 for Information	1.0.0	
Jan 2004	--	--	--	--	Added The W3C WSDL representation of the APIs specified in the present document is contained in a set of files which accompany the present document: px0326rpcenc.zip px0326rpclit.zip	1.0.1	
Jun 2004	CN_24	NP-040274	--	--	Split into multi-part specification. 29.199-0n, for n=1,2…9. Submitted to CN#24 for Information	1.0.3	
Sep 2004	CN_25	NP-040360	--	--	Draft v200 submitted to TSG CN#25 for Approval.	2.0.0	6.0.0

CR-Form-v7.1

CHANGE REQUEST

⌘ **29.199-03** **CR** **0002** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network ☒

Title:	⌘	Add display name data
Source:	⌘	CT5 John-Luc Bakker (Telcordia)
Work item code:	⌘	OSA3
Date:	⌘	09/05/2005
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	Parlay X Call Notification Web Services applications cannot not know the caller display name, when available
Summary of change:	⌘	Addition of caller name argument
Consequences if not approved:	⌘	Parlay X Call Notification Web Services applications would have to exploit proprietary means to access caller display name data.

Clauses affected:	⌘	8

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

8 Web Service interface definition

8.1 Interface: CallDirection

This subclause describes an initial set of capabilities in terms of message invocations, parameters and data types. The message-based invocations are:

- handleBusy.
- handleNotReachable.
- handleNoAnswer.
- handleCalledNumber.

These messages are initiated by the Call Notification Web Service (running in a Parlay X Gateway) and invoke an application Web Service(s), as a result of activity in the network. The result of the invocation of a handle<Event> operation is used as an indication on how the call should be handled in the network. The application can not keep control over the call after handling the event; every event handling is a separate occurrence.

Note that because the results of the invocations of the application Web Service(s) determine call handling in the network, the names of the methods are prefixed with 'handle', rather than 'notify'. The prefix 'notify' would imply a more asynchronous behaviour, whereas 'handle' shows the synchronous nature of these invocations.

The criteria for which the application Web Service(s) should be invoked, such as type of events (busy, answer, etc.), a URI to the Web Service and triggered addresses should be provisioned by the operator in an off-line process.

8.1.1 Operation: HandleBusy

The invocation of **handleBusy** requests the application to inform the gateway how to handle the call between two addresses, the **callingParty** and the **calledParty**, -where the **calledParty** is busy when the call is received. Optionally, the caller's name is provided. The application returns the **action**, which directs the gateway to perform one of the following actions:

- "Continue", resulting in normal handling of the busy event in the network, e.g. playing of a busy tone to the **callingParty.**
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParty** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParty** specified by the application.

Optionally, in the **action** parameter, the application can also indicate the charging information.

8.1.1.1 Input message: handleBusyRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CallingPartyName	xsd:string	It contains the name of the caller (optional)
CalledParty	xsd:anyURI	It contains the address of the called party. This party is busy

8.1.1.2 Output message: handleBusyResponse

Part name	Part type	Description
Action	Action	It indicates the action to be performed by the gateway

8.1.1.3 Referenced faults

None.

8.1.2 Operation: HandleNotReachable

The invocation of **handleNotReachable** requests the application to inform the gateway how to handle the call between two addresses, the **callingParty** and the **calledParty**, where the **calledParty** is not reachable when the call is received. Optionally, the caller's name is provided. The application returns the **action**, which directs the gateway to perform one of the following actions:

- "Continue", resulting in normal handling of the 'not reachable' event in the network, e.g. playing of a busy tone to the **callingParty.**
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParty** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParty** specified by the application.

Optionally, in the **action** parameter, the application can also indicate the charging information.

8.1.2.1 Input message: handleNotReachableRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CallingPartyName	xsd:string	It contains the name of the caller (optional)
CalledParty	xsd:anyURI	It contains the address of the called party. This party is not reachable

8.1.2.2 Output message: handleNotReachableResponse

Part name	Part type	Description
Action	Action	It indicates the action to be performed by the gateway

8.1.2.3 Referenced faults

None.

8.1.3 Operation: HandleNoAnswer

The invocation of **handleNoAnswer** requests the application to inform the gateway how to handle the call between two addresses, the **callingParty** and the **calledParty**, where the **calledParty** does not answer the received call. Optionally, the caller's name is provided. The application returns the **action**, which directs the gateway to perform one of the following actions:

- "Continue", resulting in normal handling of the 'no answer' event in the network, e.g. playing of a busy tone to the **callingParty.**
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParty** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParty** specified by the application.

Optionally, in the **action** parameter, the application can also indicate the charging information.

8.1.3.1 Input message: handleNoAnswerRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CallingPartyName	xsd:string	It contains the name of the caller (optional)
CalledParty	xsd:anyURI	It contains the address of the called party. This party does not answer the call

8.1.3.2 Output message: handleNoAnswerResponse

Part name	Part type	Description
Action	Action	It indicates the action to be performed by the gateway

8.1.3.3 Referenced faults

None.

8.1.4 Operation: HandleCalledNumber

The invocation of **handleCalledNumber** requests the application to inform the gateway how to handle the call between two addresses, the **callingParty** and the **calledParty**. The method is invoked when the **callingParty** tries to call the **calledParty**, but before the network routes the call to the **calledParty**. For example, the **calledParty** does not have to refer to a real end user, i.e., it could be a service number. Optionally, the caller's name is provided. The application returns the **action**, which directs the gateway to perform one of the following actions:

- "Continue", resulting in normal handling in the network, i.e. the call will be routed to the **calledParty** number, as originally dialled.
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParty** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParty** specified by the application.

Optionally, in the **action** parameter, the application can also indicate the charging information.

8.1.4.1 Input message: handleCalledNumberRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CallingPartyName	xsd:string	It contains the name of the caller (optional)
CalledParty	xsd:anyURI	It contains the address of the called party

8.1.4.2 Output message: handleCalledNumberResponse

Part name	Part type	Description
Action	Action	It indicates the action to be performed by the gateway

8.1.4.3 Referenced faults

None.

8.2 Interface: CallNotification

When call events occur in the network, the application may be notified of these events. The application does not have the ability to influence the call, as call processing continues.

Notifications are provided for call attempt, busy, not reachable and no answer events.

8.2.1 Operation: NotifyBusy

A busy notification informs the application that a call between two parties was attempted, but the called party was busy.

8.2.1.1 Input message: NotifyBusyRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CallingPartyName	xsd:string	It contains the name of the caller (optional)
CalledParty	xsd:anyURI	It contains the address of the called party. This party is busy

8.2.1.2 Output message: NotifyBusyResponse

Part name	Part type	Description
None		

8.2.1.3 Referenced faults

None.

8.2.2 Operation: NotifyNotReachable

A not reachable notification informs the application that a call between two parties was attempted, but the called party was not reachable.

8.2.2.1 Input message: NotifyNotReachableRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CallingPartyName	xsd:string	It contains the name of the caller (optional)
CalledParty	xsd:anyURI	It contains the address of the called party. This party is not reachable

8.2.2.2 Output message: NotifyNotReachableResponse

Part name	Part type	Description
None		

8.2.2.3 Referenced faults

None.

8.2.3 Operation: NotifyNoAnswer

A no answer notification informs the application that a call between two parties was attempted, but the called party did not answer.

8.2.3.1 Input message: NotifyNoAnswerRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CallingPartyName	xsd:string	It contains the name of the caller (optional)
CalledParty	xsd:anyURI	It contains the address of the called party. This party did not answer

8.2.3.2 Output message: NotifyNoAnswerResponse

Part name	Part type	Description
None		

8.2.3.3 Referenced faults

None.

8.2.4 Operation: NotifyCalledNumber

A called number notification informs the application that a call between two parties is being attempted.

8.2.4.1 Input message: NotifyCalledNumberRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CallingPartyName	xsd:string	It contains the name of the caller (optional)
CalledParty	xsd:anyURI	It contains the address of the called party

8.2.4.2 Output message: NotifyCalledNumberResponse

Part name	Part type	Description
None		

8.2.4.3 Referenced faults

None.

Annex B (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
Dec 2003	CN_21	NP-030552	--	--	Submitted to CN#22 for Information	1.0.0	
Jan 2004	--	--	--	--	Added The W3C WSDL representation of the APIs specified in the present document is contained in a set of files which accompany the present document: px0326rpcenc.zip px0326rpclit.zip	1.0.1	
Jun 2004	CN_24	NP-040274	--	--	Split into multi-part specification. 29.199-0n, for n=1,2…9. Submitted to CN#24 for Information	1.0.3	
Sep 2004	CN_25	NP-040360	--	--	Draft v200 submitted to TSG CN#25 for Approval.	2.0.0	6.0.0