

**Open Service Access (OSA);
Mapping of Parlay X Web Services to Parlay/OSA APIs;
Part 11: Audio Call Mapping;
Sub-part 2: Mapping to Multi-Party Call Control
and User Interaction**



Reference

DTR/TISPAN-01021-11-02-OSA

Keywords

API, OSA, service

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2005.

© The Parlay Group 2005.

All rights reserved.

DECT™, **PLUGTESTS™** and **UMTS™** are Trade Marks of ETSI registered for the benefit of its Members.
TIPHON™ and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Contents

Intellectual Property Rights	4
Foreword.....	4
1 Scope	5
2 References	5
3 Definitions and abbreviations.....	5
3.1 Definitions	5
3.2 Abbreviations	5
4 Mapping description.....	5
5 Sequence diagrams	6
5.1 Play message and get message play status	6
5.2 Abandon playing of message.....	7
6 Detailed mapping information.....	8
6.1 Operations	8
6.1.1 playTextMessage, playAudioMessage, playVoiceXmlMessage	8
6.1.1.1 Mapping to IpMultiPartyCallControlManager.createCall	8
6.1.1.2 Mapping to IpMultiPartyCall.createAndRouteCallLegReq	8
6.1.1.2.1 Alternative Mapping to IpMultiPartyCall.createCallLeg	9
6.1.1.2.2 Alternative Mapping to IpCallLeg.eventReportReq	9
6.1.1.2.3 Alternative Mapping to IpCallLeg.routeReq.....	10
6.1.1.3 Mapping to IpUIManager.createUICall	10
6.1.1.4 Mapping to IpUICall.sendInfoReq	10
6.1.1.5 Mapping of text	11
6.1.1.6 Mapping of voiceXmlUrl	11
6.1.1.7 Mapping of audioURL	11
6.1.2 getMessageStatus.....	12
6.1.3 endMessage.....	12
6.2 Exceptions	12
7 Additional notes	12
History	13

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN).

The present document is part 11, sub-part 2 of a multi-part deliverable covering Open Service Access (OSA); Mapping of Parlay X Web Services to Parlay/OSA APIs, as identified below:

- Part 1: "Common Mapping";
- Part 2: "Third Party Call Mapping";
- Part 3: "Call Notification Mapping";
- Part 4: "Short Messaging Mapping";
- Part 5: "Multimedia Messaging Mapping";
- Part 6: "Payment Mapping";
- Part 7: "Account Management Mapping";
- Part 8: "Terminal Status Mapping";
- Part 9: "Terminal Location Mapping";
- Part 10: "Call Handling Mapping";
- Part 11: "Audio Call Mapping";**
 - Sub-part 1: "Mapping to Generic Call Control and User Interaction";
 - Sub-part 2: "Mapping to Multi-Party Call Control and User Interaction";**
- Part 12: "Multimedia Conference Mapping";
- Part 14: "Presence Mapping".

NOTE: Part 13 has not been provided as there is currently no defined mapping between ES 202 391-13 [4] and the Parlay/OSA APIs. If a mapping is developed, it will become part 13 of this series.

The present document has been defined jointly between ETSI, The Parlay Group (<http://www.parlay.org>) and the 3GPP.

1 Scope

The present document specifies the mapping of the Parlay X Audio Call Web Service to the Multi-Party Call Control and User Interaction Service Capability Features (SCFs).

The Parlay X Web Services provide powerful yet simple, highly abstracted, imaginative, telecommunications functions that application developers and the IT community can both quickly comprehend and use to generate new, innovative applications.

The Open Service Access (OSA) specifications define an architecture that enables application developers to make use of network functionality through an open standardized interface, i.e. the Parlay/OSA APIs.

2 References

For the purposes of this Technical Report (TR), the following references apply:

[1] ETSI TR 121 905: "Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

[2] W3C Recommendation (2 May 2001): "XML Schema Part 2: Datatypes".

NOTE: Available at <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>.

[3] ETSI TR 102 397-1: "Open Service Access (OSA); Mapping of Parlay X Web Services to Parlay/OSA APIs; Part 1: Common Mapping".

[4] ETSI ES 202 391-13: "Open Service Access (OSA); Parlay X Web Services; Part 13: Address List Management".

[5] ISO 639: "Codes for the representation of names of languages".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 102 397-1 [3] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 102 397-1 [3] apply.

4 Mapping description

The Audio Call capability can be implemented with Parlay/OSA Multi-Party Call Control and User Interaction.

It is applicable to ETSI OSA 1.x/2.x/3.x, Parlay/OSA 3.x/4.x/5.x and 3GPP Releases 4 to 6.

5 Sequence diagrams

5.1 Play message and get message play status

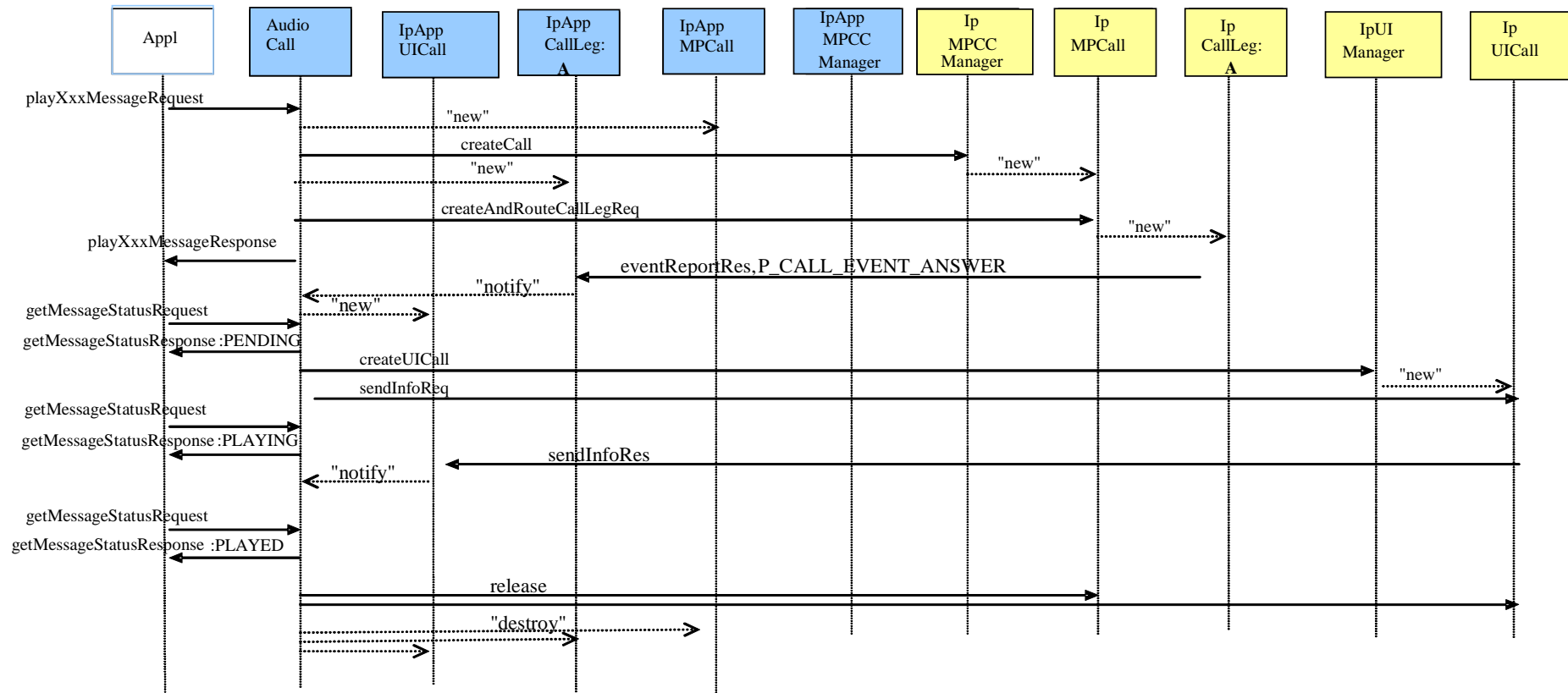


Figure 1

5.2 Abandon playing of message

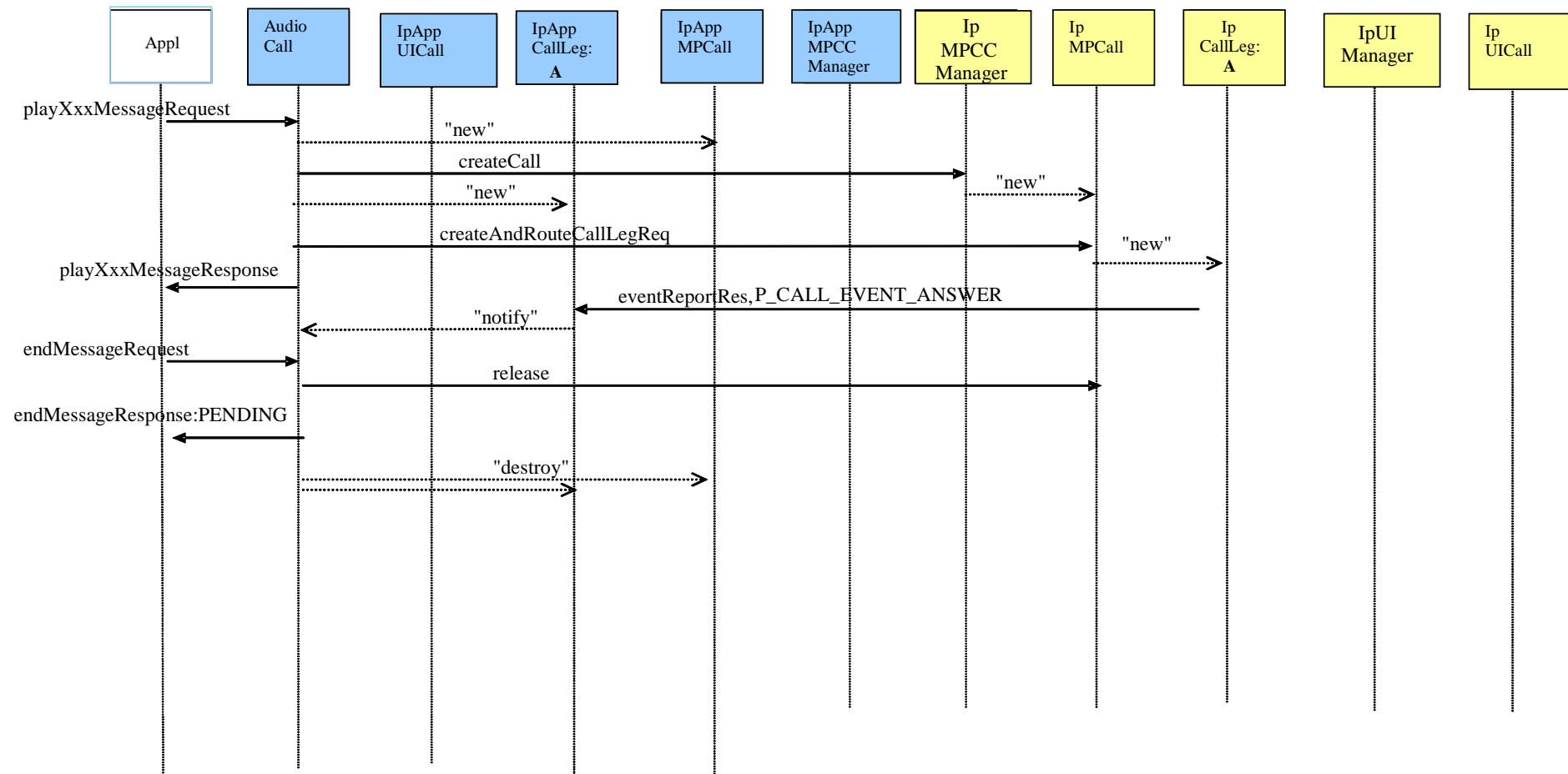


Figure 2

6 Detailed mapping information

6.1 Operations

6.1.1 playTextMessage, playAudioMessage, playVoiceXmlMessage

These operations interact with a call control service to establish a call, where the call is associated with a user interaction service context in which the audio is generated on the call. The sequence diagram in clause 5.1 illustrates the flow for these operations.

The **playXxxMessage** operation is synchronous from the Parlay X client's point of view. It is mapped to the following Parlay/OSA methods:

- `IpMultiPartyCallControlManager.createCall`
- `IpMultiPartyCall.createAndRouteCallLegReq`; OR

```
{ IpMultiPartyCall.createCallLeg,
  IpCallLeg.eventReportReq, IpCallLeg.routeReq };
```
- `IpUIManager.createUICall`;
- `IpUICall.sendInfoReq`.

6.1.1.1 Mapping to `IpMultiPartyCallControlManager.createCall`

The `IpMultiPartyCallControlManager.createCall` method is invoked with the following parameters.

Name	Type	Comment
appCall	IpAppMultiPartyCallRef	Reference to callback (internal).

The result from `IpMultiPartyCallControlManager.createCall` is of type `TpMultiPartyCallIdentifier` and is used internally to correlate the callbacks. Specifically it is correlated with the value of the **correlator** part returned to the application in the **playXxxMessageResponse** message

Parlay exceptions thrown by `IpMultiPartyCallControlManager.createCall` are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.1.2 Mapping to `IpMultiPartyCall.createAndRouteCallLegReq`

The `IpMultiPartyCall.createAndRouteCallLegReq` method is invoked with the following parameters.

Name	Type	Comment
callSessionID	TpSessionID	Not mapped. [The value provide in the result from <code>IpMultiPartyCallControlManager.createCall</code>].
eventsRequested	TpCallEventRequestSet	Not mapped. [Requests call-related event reports: i.e. including Answer, Busy, No Answer, Not Reachable.]
targetAddress	TpAddress	Specifies the destination leg to which the call should be routed. It is constructed based on the URI provided in the address part of playXxxMessageRequest , mapped as described in TR 102 397-1 [3].
originatingAddress	TpAddress	Not mapped. [Specifies the calling party leg.]
appInfo	TpCallAppInfoSet	Not mapped.
appLegInterface	IpAppCallLegRef	Not mapped. [Specifies a reference to the application interface that implements the callback interface for the new call leg. Requested events will be reported by the <code>eventReportRes()</code> operation on this interface.]

The result from `IpMultiPartyCall.createAndRouteCallLegReq` is of type `TpCallLegIdentifier` and is used internally to correlate the callbacks. It is not mapped to the Parlay X interface.

Parlay exceptions thrown by `IpMultiPartyCall.createAndRouteCallLegReq` are mapped to Parlay X exceptions as defined in clause 6.2.

An alternative to mapping to the `IpMultiPartyCall.createAndRouteCallLegReq` convenience method is a mapping to the following discrete method invocations:

- `IpMultiPartyCall.createCallLeg;`
- `IpCallLeg.eventReportReq;`
- `IpCallLeg.routeReq.`

6.1.1.2.1 Alternative Mapping to `IpMultiPartyCall.createCallLeg`

The `IpMultiPartyCall.createCallLeg` method is invoked with the following parameters.

Name	Type	Comment
<code>callSessionID</code>	<code>TpSessionID</code>	Not mapped: the result from the invocation of <code>IpMultiPartyCallControlManager.createCall</code> , as described in clause 6.1.1.1.
<code>appCallLeg</code>	<code>IpAppCallLegRef</code>	Not mapped: [Specifies a reference to the application interface that implements the callback interface for the new call leg. Requested events will be reported by the <code>eventReportRes()</code> operation on this interface.]

The result from `IpMultiPartyCall.createCallLeg` is of type `TpCallLegIdentifier` and is not mapped to the Parlay X interface.

Parlay exceptions thrown by `IpMultiPartyCall.createCallLeg` are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.1.2.2 Alternative Mapping to `IpCallLeg.eventReportReq`

The `IpCallLeg.eventReportReq` method is invoked with the following parameters.

Name	Type	Comment
<code>callLegSessionID</code>	<code>TpSessionID</code>	Not mapped: the result returned from the invocation of <code>IpMultiPartyCall.createCallLeg</code> , as described in clause 6.1.1.2.1.
<code>eventsRequested</code>	<code>TpCallEventRequestSet</code>	Not mapped. [Requests call-related event reports: i.e. including Answer, Busy, No Answer, Not Reachable.]

Parlay exceptions thrown by `IpCallLeg.eventReportReq` are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.1.2.3 Alternative Mapping to `IpCallLeg.routeReq`

The `IpCallLeg.routeReq` method is invoked with the following parameters.

Name	Type	Comment
<code>callLegSessionID</code>	<code>TpSessionID</code>	Not mapped: the result returned from the invocation of <code>IpMultiPartyCall.createCallLeg</code> , as described in clause 6.1.1.2.1.
<code>targetAddress</code>	<code>TpAddress</code>	Specifies the destination leg to which the call should be routed. It is constructed based on the URI provided in the address part of playXxxMessageRequest , mapped as described in TR 102 397-1 [3].
<code>originatingAddress</code>	<code>TpAddress</code>	Not mapped. [Specifies the calling party leg.]
<code>applInfo</code>	<code>TpCallAppInfoSet</code>	Not mapped.
<code>connectionProperties</code>	<code>TpCallLegConnectionProperties</code>	Not mapped. Specifies the properties of the connection: i.e. <code>AttachMechanism = P_CALLLEG_ATTACH_IMPLICITLY</code> .

Parlay exceptions thrown by `IpCallLeg.routeReq` are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.1.3 Mapping to `IpUIManager.createUICall`

The `IpUIManager.createUICall` method is invoked with the following parameters.

Name	Type	Comment
<code>appUI</code>	<code>IpAppUICallRef</code>	Not mapped: reference to callback (internal)
<code>uiTargetObject</code>	<code>TpUITargetObject</code>	Not mapped. [The value provide in the result from <code>IpMultiPartyCallControlManager.createCall</code>]

The result from `IpUIManager.createUICall` is of type `TpUICallIdentifier` and is used internally to correlate the callbacks. Specifically it is correlated with the value of the **correlator** part returned to the application in the **playXxxMessageResponse** message

Parlay exceptions thrown by `IpUIManager.createUICall` are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.1.4 Mapping to `IpUICall.sendInfoReq`

The `IpUICall.sendInfoReq` method is invoked with the following parameters.

Name	Type	Comment
<code>userInteractionSessionID</code>	<code>TpSessionID</code>	Not mapped: reference to callback (internal).
<code>info</code>	<code>TpUIInfo</code>	For
<code>variableInfo</code>	<code>TpUIVariableInfoSet</code>	The mapping from the text part is described in clause 6.1.1.5 The mapping from the voiceXmlUrl part is described in clause 6.1.1.6 The mapping from the audioURL part is described in clause 6.1.1.7 Some mapping support for the optional charging part: i.e. it could be mapped to a <code>VariablePartPrice</code> element(s) of the <code>variableInfo</code> parameter.
<code>language</code>	<code>TpLanguage</code>	The language part is mapped to the <code>language</code> parameter; both part and parameter conform to ISO 639 [5].
<code>repeatIndicator</code>	<code>TpInt32</code>	Not mapped.
<code>responseRequested</code>	<code>TpUIResponseRequest</code>	Not mapped. Set to <code>P_UI_RESPONSE_REQUIRED</code> .

The result from `IpUICall.sendInfoReq` is of type `TpAssignmentID` and is used internally to correlate the callbacks (e.g. invocation of `IpAppUICall.sendInfoRes/Err`). Specifically it is correlated with the value of the **correlator** part returned to the application in the **playXxxMessageResponse** message.

Parlay exceptions thrown by `IpUICall.sendInfoReq` are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.1.5 Mapping of **text**

The **text** part is of type **xsd:string** and represents the text to process and play through a Text-To-Speech engine. It is mapped to the `info` and `variableInfo` parameters as follows:

- For ETSI OSA 1.x, Parlay/OSA 3.x and 3GPP Release 4.x and subsequent releases, the **text** part is mapped to `InfoData (info.P_UI_INFO_DATA)`, which defines the data to be sent to an end-user's terminal. The data is free-format and the encoding is depending on the resources being used.
 - The Audio Call web service needs to indicate that text-to-speech processing is required from a network resource. Options for indicating this are vendor-specific.
 - One option is to include an indicator in the `InfoData` parameter: e.g. by prefixing the value of the **text** part.
 - Another option is to use the `variableInfo` parameter: e.g. the `VariablePartInteger` or `VariablePartAddress` element.
- For ETSI OSA 3.x, Parlay/OSA 5.x and 3GPP Release 6.x, an alternative mapping of the **text** part is to `InfoSynthData (info.P_UI_INFO_SYNTHESIS)`, which describes the content and how the speech synthesis will be done. Specifically the **text** part is mapped to the `InfoSynthData.TextData` field. There is no mapping to the other fields of `InfoSynthData` that define how the synthesis should be done; these fields are provisioned by the vendor.

6.1.1.6 Mapping of **voiceXmlUrl**

The **voiceXmlUrl** part is of type **xsd:anyURI** and represents the location of VoiceXML to be processed by a VoiceXML browser. It is mapped to the `info` and `variableInfo` parameters as follows:

- For ETSI OSA 1.x, Parlay/OSA 3.x and 3GPP Release 4.x and subsequent releases, the **voiceXmlUrl** part is mapped to `InfoAddress (info.P_UI_INFO_ADDRESS)`, which defines the URL of the stream to be sent to an end-user's terminal.

NOTE: In later releases of the API, the scope of the `InfoAddress` parameter is expanded to represent the URL of a voice application script or stream to be either sent to an end-user's terminal or invoked in the network in order to carry out the interaction dialogue. However an alternative parameter mapping is also available in later API releases, as described below.

- The Audio Call web service needs to indicate that VoiceXML browser processing is required from a network resource. Options for indicating this are vendor-specific.
 - One option is to provide an indicator in the `variableInfo` parameter: e.g. the `VariablePartInteger` or `VariablePartAddress` element.
- For ETSI OSA 3.x, Parlay/OSA 5.x and 3GPP Release 6.x, an alternative mapping of the **voiceXmlUrl** part is to `InfoVXMLData (info.P_UI_INFO_VXML)`, which defines the `TpString` that describes the VXML (Voice XML) page that is sent to the server for execution and interaction with the end-user. (See <http://www.w3.org/TR/2000/NOTE-voicexml-20000505/> for more information.)

6.1.1.7 Mapping of **audioURL**

The **audioUrl** part is of type **xsd:anyURI** and represents the location of audio content (WAV or MP3 file) to be played by an audio processor. It is mapped to the `info` and `variableInfo` parameters as follows:

- For ETSI OSA 1.x, Parlay/OSA 3.x and 3GPP Release 4.x and subsequent releases, the **audioUrl** part is mapped to `InfoAddress (info.P_UI_INFO_ADDRESS)`, which defines the URL of the stream to be sent to an end-user's terminal.

NOTE: In later releases of the API, the scope of the `InfoAddress` parameter is expanded to represent the URL of a stream to be either sent to an end-user's terminal or invoked in the network in order to carry out the interaction dialogue. However an alternative parameter mapping is also available in later API releases, as described below.

- The Audio Call web service needs to indicate that audio processing is required from a network resource. Options for indicating this are vendor-specific.
 - One option is to provide an indicator in the `variableInfo` parameter: e.g. the `VariablePartInteger` or `VariablePartAddress` element.
- For ETSI OSA 2.x, Parlay/OSA 4.x and 3GPP Release 5.x and subsequent releases, an alternative mapping of the **audioUrl** part is to `InfoWaveData` (`info.P_UI_INFO_WAVE`) or `InfoAuData` (`info.P_UI_INFO_AU`), which defines the WAVE or AU data to be sent to an end-user's terminal. Both these elements are of type `TpOctetSet` and should contain the URL value of the **audioUrl** part. If this is not possible, or if other audio formats are required (e.g. MP3 or others, as specified in the **AudioFormatsSupported** service policy), then the `variableInfo` parameter can also be used.

6.1.2 getMessageStatus

The message status is managed by the Web Service implementation, reflecting the current actions that have been taken by the services and interactions with the call control and user interaction services. The sequence diagram in clause 5.1 illustrates the flow for this operation. The value of the **result** part of the **getMessageStatusResponse** message reflects the current stage of processing that was initiated by the **playXxxMessage** operation: i.e. **Pending**, **Playing** and **Played**. A fourth value, **Error**, is returned instead, if an error is reported by the Parlay/OSA interface that results in the premature termination of processing associated with the **playXxxMessage** operation: e.g. invocation of `IpAppCallLeg.eventReportErr`, `IpAppUICall.sendInfoErr`, method exceptions, etc.

6.1.3 endMessage

This operation terminates the call and associated user interaction session, using the release operation on the user interaction and call objects respectively. The sequence diagram in clause 5.2 illustrates the flow for this operation.

The **endMessage** operation is synchronous from the Parlay X client's point of view. It is mapped to the Parlay/OSA method: `IpMultiPartyCall.release`.

The `IpMultiPartyCall.release` method is invoked with the following parameters.

Name	Type	Comment
<code>callSessionID</code>	<code>TpSessionID</code>	Not mapped. [The value provide in the result from <code>IpMultiPartyCallControlManager.createCall</code>].
<code>cause</code>	<code>TpReleaseCause</code>	Assigned a value indicating application-initiated call termination.

Parlay exceptions thrown by `IpMultiPartyCall.release` are mapped to Parlay X exceptions as defined in clause 6.2.

6.2 Exceptions

In addition to the common mapping of Parlay/OSA API method exceptions to Parlay X Web Service exceptions, which is defined in TR 102 397-1 [3], there are the following service-specific exception mappings:

Parlay/OSA Exception	Service Exception	Notes
<code>P_ILLEGAL_ID</code>	SVC0002	
<code>P_ID_NOT_FOUND</code>	SVC0001	With error number.

7 Additional notes

No additional notes.

History

Document history		
V1.1.1	August 2005	Publication