3GPP TSG CN Plenary Meeting #25 **NP-040357**
08-10 September 2004, Palm Springs, CA, USA

Source: **CN5 (OSA)**

Title: **3 Rel-6 CR 29.198-05**

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

Document for: **APPROVAL**

Doc-1st-	Spec	CR	R	Phase	Subject	Cat	VCur	Doc-2nd	WI
NP-040357	29.198-05	054	--	Rel-6	Add getMessageListReq() within the IpUIAdminManager interface	B	6.1.0	N5-040587	OSA3
NP-040357	29.198-05	055	--	Rel-6	Change description of InfoAddress within TpUIInfo	D	6.1.0	N5-040588	OSA3
NP-040357	29.198-05	056	--	Rel-6	Changes to the TpUIRecognitionGrammer parameter	F	6.1.0	N5-040593	OSA3

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-05 CR 054 ⌘rev - ⌘ Current version: 6.1.0 ⌘

*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:	⌘	Add getMessageListReq() within the IpUIAdminManager interface
Source:	⌘	CN5 (richard.stretch@bt.com)
Work item code:	⌘	OSA3
Date:	⌘	12/08/2004
Category:	⌘	**B**
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:

2	*(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)*

Reason for change:	⌘	The IpUIAdminManager/IpAppUIAdminManager interfaces were introduced in order to enable the OSA application to manage the messages it has available to it. In order to make the message management feature more efficient it is proposed to add an additional method (getMessageListReq()) which will allow the application to obtain a list of the current messages which it has available on the underlying network resource. In addition to the new method a new data type definition is required (which can be used to return the list of messageIDs.
Summary of change:	⌘	This change adds an additional method "getMessageListReq()" to the IpUIAdminManager interface to enable applications to search for the messageIDs of those messages it has provisioned or recorded. Corresponding getMessageListRes() and getMessageListErr() methods have been added to the IpAppUIAdminManager interface.
Consequences if not approved:	⌘	Ensures applications are able to manage their messages.

Clauses affected:	⌘	8.2, 11 (new)

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

Other comments: ⌘

How to create CRs using this form:

8.2 Generic User Interaction Administration Interface Classes

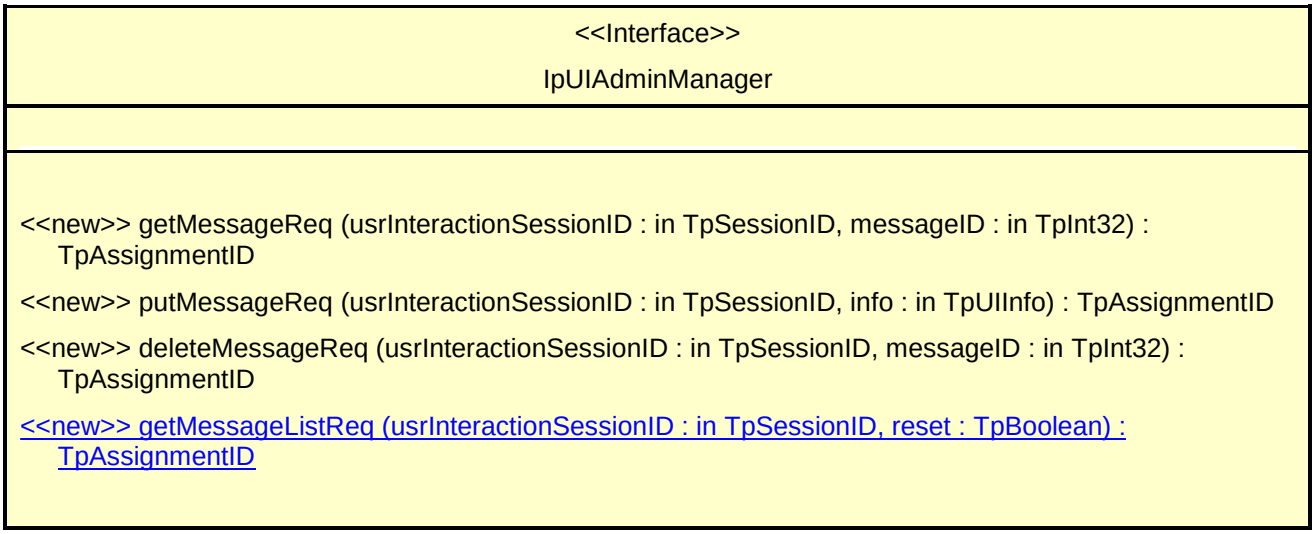
Interface Class IpUIAdminManager

Inherits from: IpService.

The Generic User Interaction Administration Manager Service interface is used by applications to manage user announcement and recorded messages on the gateway. This Service is represented by the IpUIAdminManager interface that interfaces to the service provided by the network. To handle responses and reports, the developer must implement IpAppUIAdminManager interface to provide the callback mechanism.

The application context will ensure that one application doesn't interfere with the messages of another application.

The User Interaction Administration Manager Service Interface provides functions to manage the messages.



8.2.1.1 Method <<new>> getMessageReq()

This asynchronous method allows retrieving the user announcement or recorded message content from the gateway.

Returns: assignmentID

Specifies the ID assigned by the user interaction interface for a user interaction request.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

messageID : in TpInt32

Specifies the message ID.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE, P_ILLEGAL_ID, P_ID_NOT_FOUND

8.2.1.2 Method <<new>> putMessageReq()

This asynchronous method allows putting a user announcement message content onto the gateway. The gateway will allocate the messageID and return it to the application on the putMessageRes() confirmation.

Returns: assignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

info : in TpUIInfo

Specifies the information to send to the user. This information can be either an ID (for pre-defined announcement or text), a text string, or an URL (indicating the information to be sent, e.g. an audio stream).

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_ILLEGAL_ID, P_ID_NOT_FOUND

8.2.1.3 Method <<new>> deleteMessageReq()

This asynchronous method allows deleting a user announcement or recorded message.

Returns: assignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

messageID : in TpInt32

Specifies the message ID.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_ILLEGAL_ID, P_ID_NOT_FOUND

8.2.1.4 Method <<new>> getMessageListReq ()

This synchronous method allows the application to retrieve a list of Message Ids for all its recorded messages or user announcements.

Returns: assignmentID

Specifies the ID assigned by the user interaction admin manager interface in order to correlate the response.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

reset : in TpBoolean

TRUE: Indicates that the application intends to obtain the list of messages starting from the beginning.

FALSE: Indicates that the application requests the next part of the list that has not (yet) been obtained since the last call to this method with this parameter set to TRUE.

The first time this method is invoked, reset shall be set to TRUE. Following the receipt of a final indication in the getMessageListRes(), for the next call to this method reset shall be set to TRUE. P_TASK_REFUSED may be thrown if these conditions are not met.

Returns

TpAssignmentID

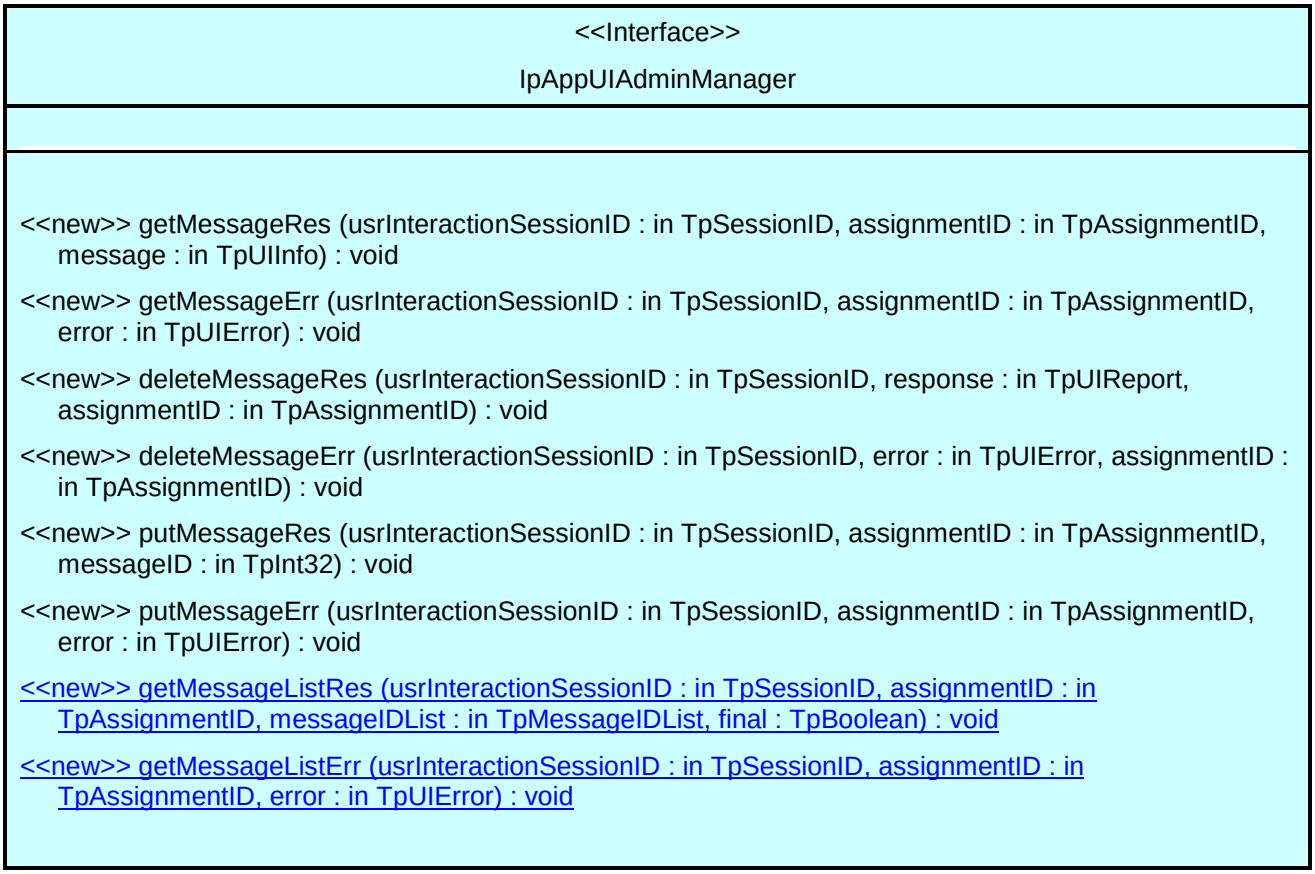
Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE.

8.2.2 Interface Class IpAppUIAdminManager

Inherits from: IpInterface.

The User Interaction Administration Manager Application Interface is implemented by the client application and is used to handle administration user interaction request responses and reports.



8.2.2.1 Method <<new>> getMessageRes()

This method returns the message content if the message was retrieved successfully.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the user interaction interface for a user interaction request.

message : in TpUIInfo

Specifies the UI Information containing the message content information.

8.2.2.2 Method <<new>> getMessageErr()

This method indicates that the request to retrieve a message was not successful.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the user interaction interface for a user interaction request.

error : in TpUIError

Specifies the error which led to the original request failing.

8.2.2.3 Method <<new>> deleteMessageRes()

This method indicates that the request to delete a message was successful.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

response : in TpUIReport

Specifies the type of response received from the device where the message was stored.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the user interaction interface for a user interaction request.

8.2.2.4 Method <<new>> deleteMessageErr()

This method indicates that the request to delete a message was not successful.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

error : in TpUIError

Specifies the error which led to the original request failing.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the user interaction interface for a user interaction request.

8.2.2.5 Method <<new>> putMessageRes()

This asynchronous method confirms that the request to put the message content was successful.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the user interaction interface for a user interaction request.

messageID : in TpInt32

Specifies the message ID that was allocated by the gateway.

8.2.2.6 Method <<new>> putMessageErr()

This asynchronous method indicates that the request to put the message content resulted in an error.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the user interaction interface for a user interaction request.

error : in TpUIError

Specifies the error which led to the original request failing.

8.2.2.7 Method <<new>> getMessageListRes()

This asynchronous method returns the result of a getMessageListReq() method. Whether there are still more messages that can be listed yet will be indicated with the final parameter.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the user interaction admin manager interface in order to correlate with the request.

messageIDList : in TpMessageIDList

Specifies the list of MessageIDs returned by the SCF.

final : in TpBoolean

Indication whether the returned list is the final part of the complete list (TRUE) or if there are still parts of the list to retrieve (FALSE).

8.2.2.8 Method <<new>> getMessageListErr()

This asynchronous method indicates that the request to list the messageIDs was not successful.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the user interaction admin manager interface in order to correlate with the request.

error : in TpUIError

Specifies the error which led to the original request failing.

==========

11.42 TpMessageIDList

This data type defines a Numbered List of Data Elements of type TpInt32.

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-05** **CR** **055** ⌘ **rev** **-** ⌘ Current version: **6.1.0** ⌘

*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:	⌘	Change description of InfoAddress within TpUIInfo
Source:	⌘	CN5 (richard.stretch@bt.com)
Work item code:	⌘	OSA3
Date:	⌘	12/08/2004
Category:	⌘	**D**
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:

2	*(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)*

Reason for change:	⌘	The description of InfoAddress withinTpUIInfo does not provide sufficient enough information as to how the element can be used to define a URL for VXML specific use, without the need to utilize P_UI_INFO_VXML within TPUIInfoType.
Summary of change:	⌘	The proposal is therefore to provide additional clarification of InfoAddress in TpUIInfo to explicitly show that the URL may reference voice application script.
Consequences if not approved:	⌘	May lead the reader of the spec to assume that only P_UI_INFO_VXML within TPUIInfoType can be used to exchange information concerning voice application scripts

Clauses affected:	⌘	11.7

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

Other comments:	⌘	

How to create CRs using this form:

11.17 TpUIInfo

Defines the Tagged Choice of Data Elements that specify the information to send to the user.

	Tag Element Type	
	TpUIInfoType	

Tag Element Value	Choice Element Type	Choice Element Name
P_UI_INFO_ID	TpInt32	InfoId
P_UI_INFO_DATA	TpString	InfoData
P_UI_INFO_ADDRESS	TpURL	InfoAddress
P_UI_INFO_BIN_DATA	TpOctetSet	InfoBinData
P_UI_INFO_UUENCODED	TpString	InfoUUEncData
P_UI_INFO_MIME	TpOctetSet	InfoMimeData
P_UI_INFO_WAVE	TpOctetSet	InfoWaveData
P_UI_INFO_AU	TpOctetSet	InfoAuData
P_UI_INFO_VXML	TpString	InfoVXMLData
P_UI_INFO_SYNTHESIS	TpUISynthesisInfoData	InfoSynthData

The choice elements represent the following:

InfoID: defines the ID of the user information script or stream to send to an end-user. The values of this data type are operator specific.

InfoData: 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..

InfoAddress: defines the URL of the text, 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.

InfoBinData: defines the binary data to be sent to an end-user's terminal. The data is a free-format, 8-bit quantity that is guaranteed not to undergo any conversion when transmitted.

InfoUUEncData: defines the UUEncoded data to be sent to an end-user's terminal.

InfoMimeData: defines the MIME data to be sent to an end-user's terminal.

InfoWaveData: defines the WAVE data to be sent to an end-user's terminal.

InfoAuData: defines the AU data to be sent to an end-user's terminal.

InfoVXMLData: 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.

InfoSynthData: defines the TpUISynthesisInfoData that describes the content and how the speech synthesis will be done.

InfoSynthData allows the application to utilize the fundamental speech synthesis capabilities of the server without dependency VXML, while InfoVXMLData allows the application to send a complex VXML program (including call control, flow control, dynamic content, menuing, etc) to the server for execution with little change to the OSA application itself.

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-05 CR 056 ⌘rev - ⌘ Current version: 6.1.0 ⌘

*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:	⌘	Changes to the TpUIRecognitionGrammer parameter
Source:	⌘	CN5 (richard.stretch@bt.com)
Work item code:	⌘	OSA3
Date:	⌘	12/08/2004
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:

2	*(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)*

Reason for change: ⌘ The RecognitionCriteria field has previuously been modified to introduce support for voice recognition matching.

The key feature considered here is the Grammar parameter. This is considered below.

Modification of the Grammar parameter.

The OSA Release 6 UI specification references the Java Speech Grammar Format. Since then the W3C Voice Browser Working Group have been working on a number of W3C recommendations, which specify several aspects around the implementation of voice application development. The key recommendation is the Speech recognition Grammar Specification Version 1. Within this project the SRGS is the key document since it is referenced from both the Voice XML 2.0 and SALT 1.0 specifications. So therefore, any grammar specification in line with this recommendation would likely offer an implementation option within the majority of Next Generation speech platforms.

The proposed changes to the use of in-line grammars are simply to remove the reference to the JSGF and to simplify the in-line grammar.

Furthermore, given the complexities of the Voice XML and SALT specifications as well as the lengthy grammar standards it is recommended to modify the interface such that it offers the developer access to a few basic VXML/SALT platform capabilities.

The suggested scope is therefore to keep the interactions and the OSA application as simple as possible by limiting the UI capabilities. For simple interactions the developer should create a simple announcement and recognition interaction using the in-line grammar concept. This should be limited to a maximum word list size but could be an operator specific feature dependent on the customer's SLA.

The use of a referenced grammar file is suggested to be deleted. It is felt that the simpler in-line matching should suffice for the OSA developer's experience.

Summary of change:	⌘	To remove references to the JSGF in favour of SRGS. Simplification of the in-line grammar.
Consequences if not approved:	⌘	The present description ties the applications into Voice XML 1.0 which is an old reference and has now been superceeded.

Clauses affected:	⌘	2, 11.41			
Other specs affected:	⌘	**Y**	**N**		
		X		Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:

2 References

[1] 3GPP TS 29.198-1: "Open Service Access (OSA) Application Programming Interface (API); Part 1: Overview~~Open Service Access; Application Programming Interface; Part 1: Overview~~".

[2] 3GPP TS 22.127: "~~Stage 1~~ Service Requirement for the Open Services Access (OSA); Stage 1~~Service Requirement for the Open Service Access (OSA)~~".

[3] 3GPP TS 23.127: "Virtual Home Environment (VHE) / Open Service Access (OSA)~~Virtual Home Environment~~".

[4] Speech Recognition Grammar Specification Version 1. (http://www.w3.org/TR/2004/REC-speech-grammar-20040316/)

11.41 TpUIRecognitionGrammar

Defines a string that consists of an in-line grammar that specifies the syntax of the speech to be recognized. The format of this string is based on a subset of the Voice XML 1.0 grammar element tag. The in-line grammar text must be enclosed within <grammar> …..</grammar> element tags. The contents of the grammar specifies the allowable input that the voice recognition will accept. The Voice XML grammar specifies the set of utterances that a user may speak to perform an action and specifies the corresponding string value for the result.

The following table describes the features that provide a language for describing context-free grammars.

Feature	Purpose
word or words	(terminals, tokens) need not be quoted
[x]	optional x
(...)	Grouping
x {value text}	arbitrary value text may be associated with x
x*	0 or more occurrences of x
x+	1 or more occurrences of x
x y z ...	a sequence of x then y then z then ...
x | y | z | ...	a set of alternatives of x or y or z or ...
<rule>	rule names (non-terminals) are enclosed in <>
<rule> = x;	a private rule definition
public <rule> = x;	a public rule definition

The format of the grammar tag is:

```
<grammar~~ type="application/x-jsgf"~~> grammar content </grammar>
~~or~~

~~<grammar type="application/x-jsgf" src="url" />~~
```

The grammar defines a possible set of utterances. The text of the utterance itself is used as the value, if the value text is not explicitly specified with {value}.

This form is particularly convenient for expressing simple lists of alternative ways of saying the same thing, for example:

```
<grammar ~~type="application/x-jsgf"~~>
     [please] help [me] [please] | [please] I (need | want) help [please]
</grammar>


<grammar ~~type="application/x-jsgf"~~>
    hamburger | burger {hamburger} | (chicken [sandwich]) {chicken}
</grammar>
```

In the first example, any of the ways of saying "help" result in a valid response. In the second example, the user may say "hamburger" or "burger" and the response will be given the value "hamburger", or the user may say "chicken" or "chicken sandwich" and the result will be given the value "chicken".

If the grammar can not be matched, then a sendInfoAndCollectErr will result, with an P_IMPROPER_USER_RESPONSE.

For a better description and further examples of ~~Voice XML~~in-line grammar creation see [4].~~: version 1.0 and the Java Speech Grammar Format, see:~~