3GPP TSG CN Plenary Meeting #22 **NP-030555**
10 - 12 December 2003, Maui, Hawaii, USA

Source: **CN5 (OSA)**

Title: **Rel-6 CR 29.198-05 OSA API Part 5: Generic user interaction**

Agenda item: **9.7**

Document for: **APPROVAL**

Doc-1st-Level	Spec	CR	R	Ph	Subject	Cat	Version-Current	Doc-2nd-Lev	WI
NP-030555	29.198-05	044	-	Rel-6	Add speech recognition/synthesis capability to the Generic User Interaction	B	5.4.0	N5-030411	OSA3

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-05** **CR** **044** ⌘ **rev** **-** ⌘ Current version: **5.4.0** ⌘

*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 speech recognition/synthesis capability to the Generic User Interaction
Source:	⌘	CN5 (scottjb@us.ibm.com)
Work item code:	⌘	OSA3
Date:	⌘	18/07/2003
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 Generic User Interaction SCF interface does not adequately provide for speech recognition or synthesis, which are important features that would ideally need to be implemented on the OSA Gateway.

This description proposes a specific set of changes necessary to make these features possible.

Summary of change: ⌘ The new capabilities for speech recognition / synthesis should be integrated in a compatible way. The new capabilities should enable applications to be written without ambiguity or dependency on any particular speech engine.

This description provides for using VXML directly by the application, and also provides speech engine independent and VXML independent datatype definitions to allow the Parlay Application programmer that is not a VXML programmer to utilize these speech features as primitive capabilities, and is designed such that the Parlay API constructs can be easily mapped to VXML by the implementation.

The VXML definition provides not only speech synthesis/recoginition features, but also provides call control, flow control, menuing, and other features that are redundant with other aspects of the OSA API. This solution stresses the use of new API capabilities which allows application logic to occur in the client application, as opposed to VXML, which allows the client application to send VXML to the SCS server for execution. Both approaches are enabled in this solution.

To determine if these features are enabled by a particular implementation, standard properties could be used. Since synthesis is essentially a datatype defined in PUIInfoType the existing service property is adequate. However, for recognition, an additional property is necessary.
P_SPEECH_RECOGNITION_SUPPORTED boolean, true for supporting the new recognition features

for synthesis:

The TpUIInfo (P_UI_INFO_SYNTHESIS) type specifies that text will be sent to the gateway and speech will be synthesized according to the parameters.

```
enum TpUIInfoType {
        P_UI_INFO_ID,
        P_UI_INFO_DATA,
        P_UI_INFO_ADDRESS,
        P_UI_INFO_BIN_DATA,
        P_UI_INFO_VXML,
        P_UI_INFO_SYNTHESIS
};

union TpUIInfo switch(TpUIInfoType) {
   case P_UI_INFO_ID: TpInt32 InfoID;
   case P_UI_INFO_DATA: TpString InfoData;
   case P_UI_INFO_ADDRESS: TpURL InfoAddress;
   case P_UI_INFO_BIN_DATA: TpOctetSet InfoBinData;
   case P_UI_INFO_VXML: TpString;
   case P_UI_INFO_SYNTHESIS:
            TpUISynthensisInfoData InfoSynthData;
};
```

The characteristics of the speaker are defined with a set of types for gender, age, rate, range and pronunciation.

```
enum TpUISynthesisGender {
    P_UI_GENDER_MALE,
    P_UI_GENDER_FEMALE
};

enum TpUISynthesisAge {
    P_UI_AGE_CHILD,
    P_UI_AGE_YOUNG_ADULT,
    P_UI_AGE_ADULT,
    P_UI_AGE_OLDER_ADULT
};

typedef  TpInt32 TpUISynthesisRate;          /* the rate of the speech */

typedef TpInt32 TpUISynthesisRange ;      /*  the range of the speech */

enum TpUIPronounceType {
    P_UI_PRONOUNCE_IPA,
    P_UI_PRONOUNCE_SOUNDSLIKE
};

struct TpUIWordOverride {
    TpString Spelling;      /* the spelling of a word the synthesizer
                               has problems with */
    TpUIPronounceType PronounceType; /* the type of pronounciation
                                        to use */
    TpString PronounceAs;/* the pronounciation definition for this word */
};

typedef sequence <TpUIWordOverride>TpUIWordOverrideList;
```

The TpUIPronounceType defines the type of pronunciation definition that is being used. The International Phonetic Alphabet (IPA) representation can be used to specify pronunciations. For more information see:

http://www2.arts.gla.ac.uk/IPA/ipachart.html
http://charts.unicode.org/Unicode.charts/normal/U0250.html

Also, simple sound-alike replacements can also be used,such as “I triple E” for IEEE.

The TpUISynthesisInfoData specifies the information to use in generating the desired effects on the generated voice.

```
struct TpUISynthesisInfoData {
    TpUISynthesisGender SpeakerGender;       /* gender of speaker */
    TpUISynthesisAge SpeakerAge;             /* age of the speaker */
    TpUISynthesisRate SpeakerRate;           /*  rate of the speaker */
    TpUISynthesisRange SpeakerRange;         /* range of speech of the
                    speaker, larger lively, smaller more monotone */
    TpString TextData;                       /* text to speak */
    TpUIWordOverrideList WordOverrideList;   /* pronunciation
                                                override */
};
```

The TextData field may contain a subset of VXML tags to affect the processing of text. The <break> tag specifies a timing pause. The <emp> tag specifies emphasis on a word or phrase.

```
<break size="milliseconds"> // specifies a timing pause in the
                            // middle of the text in milliseconds
<emp>A word</emp>           // apply emphasis to a word or words
```

The speech synthesis parameters or text processing tags will be interpreted as hints and may be ignored by the speech synthesis engine.

For Speech Recognition:

The TpUICollectCriteria structure is updated to enable the voice recognition capabilities. For DTMF recognition or voice recognition, the request may ask for either DTMF or Voice Recognition to be done on the calling party's response, Therefore, both DTMF collection criteria and Voice Recognition Collection Criteria should be specified in the same request structure.

Note that additional fields have been added to the TpUICollectCriteria structure. The first of which is the collect mode, which uses the value of '0' zero to indicate DTMF only, which is the legacy mode. Although, this is a modification of the structure, existing software that does not reference the new fields, will default to the DTMF legacy mode, and the recognition criteria object will be null. Since, gateway and client application will utilize the same level of IDL, there should not be a serialization issue. If this strategy to add the extra criteria fields is not acceptable, then another sendInfoAndCollectReqExtended method can be added.

```
enum TpUICollectMode {
  P_UI_COLLECT_MODE_DTMF,    /* accept DTMF input */
  P_UI_COLLECT_MODE_VOICE,  /* accept Voice Recognition Input */
  P_UI_COLLECT_MODE_DTMFANDVOICE      /* accept both DTMF
                                         and Voice */
};

typedef TpString TpUIRecognitionSpeakerID;

typedef TpString TpUIRecognitionGrammar;

struct TpUIRecognitionProperty {
        TpString PropertyName;
        TpString PropertyValue;
};

typedef sequence <TpUIRecognitionProperty>
                                TpUIRecognitionPropertyList;

struct TpUIRecognitionCriteria {
  TpUIRecognitionSpeakerID SpeakerID;   /* ID for associating speaker
                                           with a speaker profile,
```

```
                                                    hint for better quality */
    TpUIRecognitionPropertyList Properties; /* Properties for recognition
                                                    engine */
    TpUIRecognitionGrammar Grammer;      /* The grammar of the
                                                    expected input */
};

struct TpUICollectCriteria {
    TpInt32 MinLength;              /* applies to DTMF or Recognition */
    TpInt32 MaxLength;              /* applies to DTMF or Recognition */
    TpString EndSequence;           /* applies to DTMF */
    TpDuration StartTimeout;        /* applies to DTMF or Recognition */
    TpDuration InterCharTimeout;/* applies to DTMF */
    TpUICollectMode CollectMode; /* applies to DTMF and
                                                    Recognition */
    TpUIRecognitionCriteria RecognitionCriteria; /* applies to
                                                    Recognition */
};
```

The TpUIRecognitionProperty properties are hints to the recognition engine on how it should interpret and recognize the input.

The defined properties are:

P_RECOGNITION_PROPERTY_CONFIDENCE_LEVEL - The speech recognition confidence level, a float value in the range of 0.0 to 1.0. Results are rejected (a nomatch event is thrown) when the recognitions engine's confidence in its interpretation is below this threshold. A value of 0.0 means minimum confidence is needed for a recognition, and a value of 1.0 requires maximum confidence. The default value is 0.5.

P_RECOGNITION_PROPERTY_SENSITIVITY - Set the sensitivity level. A value of 1.0 means that it is highly sensitive to quiet input. A value of 0.0 means it is least sensitive to noise. The default value is 0.5.

P_RECOGNITION_PROPERTY_SPEEDVSACCURACY - A hint specifying the desired balance between speed vs. accuracy. A value of 0.0 means fastest recognition. A value of 1.0 means best accuracy. The default is value 0.5.

P_RECOGNITION_PROPERTY_COMPLETE_TIMEOUT - The speech timeout value to use when an active grammar is matched.

The TpRecognitionGrammar string consists of an inline grammar fragment, specified by a subset of Voice XML 1.0, using the grammar element tag. The <grammar> element tag 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 "word" words	(terminals, tokens) need not be quoted
<rule>	rule names (non-terminals) are enclosed in <>
[x]	optionally x
(...)	Grouping
x {tag text}	arbitrary "tag" text may be associated with any of the above
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> = 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>
or
<grammar type="application/x-jsgf" src="url" />

The grammar idefines 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 of Voice XML version 1.0 and the Java Speech Grammar Format, see:
http://java.sun.com/products/java-media/speech/forDevelopers/JSGF/index.html.

Changes derived from:
ftp://ftp.3gpp.org/specs/2003-06/Rel-5/29_series/

Consequences if not approved:	⌘	Voice Recognition can not be done on the gateway through a standard interface.

Clauses affected:	⌘	8, 10, 11

		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:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.

8.3.1 Method sendInfoReq()

This asynchronous method plays an announcement or sends other information to the user.

Returns: assignmentID

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

Parameters

userInteractionSessionID : 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:

- an infoID, identifying pre-defined information to be sent (announcement and/or text);

- a string, defining the text to be sent;

- a URL , identifying pre-defined information or data to be sent to or downloaded into the terminal. A URL enables the application to utilize dynamic multi-media content by reference.

- Binary Data, identifying pre-defined information or data to be sent to or downloaded into the terminal. Binary data enables the application to utilize dynamic multi-media content directly.

- a VXML string defines the Voice XML page to execute on the server and interact with the end-user. The VXML page execution continues until an <exit/> tag is encountered, which results in a sendInfoRes() callback.

- a SynthesisInfo structure defines the text to synthesize and how the synthesis should be done.

language : in TpLanguage

Specifies the Language of the information to be sent to the user.

variableInfo : in TpUIVariableInfoSet

Defines the variable part of the information to send to the user.

repeatIndicator : in TpInt32

Defines how many times the information shall be sent to the end-user. A value of zero (0) indicates that the announcement shall be repeated until the call or call leg is released or an abortActionReq() is sent.

responseRequested : in TpUIResponseRequest

Specifies if a response is required from the call user interaction service, and any action the service should take.

Returns

TpAssignmentID

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_NETWORK_STATE,P_ILLEGAL_ID,P_ID_NOT_FOUND

8.3.2 Method sendInfoAndCollectReq()

This asynchronous method plays an announcement or sends other information to the user and collects some information from the user. The announcement usually prompts for a number of characters (for example, these are digits or text strings such as "YES" if the user's terminal device is a phone).

Returns: assignmentID

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

Parameters

`userInteractionSessionID : in TpSessionID`

Specifies the user interaction session ID of the user interaction.

`info : in TpUIInfo`

Specifies the ID of the information to send to the user. This information can be:

- an infoID, identifying pre-defined information to be sent (announcement and/or text);

- a string, defining the text to be sent;

- a URL , identifying pre-defined information or data to be sent to or downloaded into the terminal. A URL enables the application to utilize dynamic multi-media content by reference.

- Binary Data, identifying pre-defined information or data to be sent to or downloaded into the terminal. Binary data enables the application to utilize dynamic multi-media content directly.

- a VXML string defines the Voice XML page to execute on the server and interact with the end-user. The VXML page execution continues until an <exit/> tag is encountered, which results in a sendInfoAndCollectRes() callback with the value of the expr= attribute.

- a SynthesisInfo structure defines the text to synthesize and how the synthesis should be done.

`language : in TpLanguage`

Specifies the Language of the information to be sent to the user.

`variableInfo : in TpUIVariableInfoSet`

Defines the variable part of the information to send to the user.

`criteria : in TpUICollectCriteria`

Specifies additional properties for the collection of information, such as the maximum and minimum number of characters, end character, first character timeout and inter-character timeout. This parameter also specifies whether voice recognition would be used.

`responseRequested : in TpUIResponseRequest`

Specifies if a response is required from the call user interaction service, and any action the service should take. For this case it can especially be used to indicate e.g. the final request.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE, P_ILLEGAL_ID, P_ID_NOT_FOUND, P_ILLEGAL_RANGE, P_INVALID_COLLECTION_CRITERIA

10.1 User Interaction Service Properties

The following table lists properties relevant for the User Interaction API.

Property	Type	Description
P_INFO_TYPE	INTEGER_SET	Specifies whether the UI SCS supports text or URLs etc. Allowed value ~~set:~~ ~~(P_INFO_ID,~~ ~~P_URL,~~ ~~P_TEXT)~~are defined by TpUIInfoType.
P_SPEECH_RECOGNITION_SUPPORTED	BOOLEAN	Value: TRUE when the speech recognition features are supported

The previous table lists properties related to capabilities of the SCS itself. The following table lists properties that are used in the context of the Service Level Agreement, e.g. to restrict the access of applications to the capabilities of the SCS.

Property	Type	Description
P_TRIGGERING_ADDRESSES	ADDRESS_RANGE_SET	Specifies which numbers the notification may be set
P_SERVICE_CODE	INTEGER_SET	Specifies the service codes that may be used for notification requests.

11.9 TpUICollectCriteria

Defines the Sequence of Data Elements that specify the additional properties for the collection of information, such as the end character, first character timeout, inter-character timeout, and maximum interaction time. The CollectMode element defines the type of data that is to be collected. DTMF and Voice Recognition can be used separately or in combination. The P_SPEECH_RECOGNITION_SUPPORTED property defines whether the voice recognition features are supported.

Structure Element Name	Structure Element Type
MinLength	TpInt32
MaxLength	TpInt32
EndSequence	TpString
StartTimeout	TpDuration
InterCharTimeout	TpDuration
CollectMode	TpUICollectMode
RecognitionCriteria	TpUIRecognitionCriteria

The structure elements specify the following criteria:

MinLength: Defines the minimum number of characters (e.g. digits) to collect. Applies to DTMF collection and voice recognition.

MaxLength: Defines the maximum number of characters (e.g. digits) to collect. Applies to DTMF collection and voice recognition.

EndSequence:	Defines the character or characters which terminate an input of variable length, e.g. phone numbers. Applies to DTMF collection only.
StartTimeout:	specifies the value for the first character time-out timer. The timer is started when the announcement has been completed or has been interrupted. The user should enter the start of the response (e.g. first digit) before the timer expires. If the start of the response is not entered before the timer expires, the input is regarded to be erroneous. After receipt of the start of the response, which may be valid or invalid, the timer is stopped. Applies to DTMF collection and voice recognition.
InterCharTimeOut:	specifies the value for the inter-character time-out timer. The timer is started when a response (e.g. digit) is received, and is reset and restarted when a subsequent response is received. The responses may be valid or invalid. the announcement has been completed or has been interrupted. Applies to DTMF collection only.
CollectMode:	Defines the type of collection to do. Applies to DTMF collection and voice recognition. The default is DTMF collection only.
RecognitionCriteria:	Defines the criteria for voice recognition.

Input is considered successful if the following applies:

If the EndSequence is not present (i.e. an empty string):

- when the InterCharTimeOut timer expires; or
- when the number of valid digits received equals the MaxLength.

If the EndSequence is present:

- when the InterCharTimeOut timer expires; or
- when the EndSequence is received; or
- when the number of valid digits received equals the MaxLength.

In the case the number of valid characters received is less than the MinLength when the InterCharTimeOut timer expires or when the EndSequence is received, the input is considered erroneous.

The collected characters (including the EndSequence) are sent to the client application when input has been successful.

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 or stream to be sent to an end-user's terminal.

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.

11.18 TpUIInfoType

Defines the type of the information to be sent to the user.

Name	Value	Description
P_UI_INFO_ID	0	The information to be sent to an end-user consists of an ID.
P_UI_INFO_DATA	1	The information to be sent to an end-user consists of a data string.
P_UI_INFO_ADDRESS	2	The information to be sent to an end-user consists of a URL.
P_UI_INFO_BIN_DATA	3	The information to be sent to an end-user consists of a 8 bit binary data set.
P_UI_INFO_UUENCODED	4	The information to be sent to an end-user consists of UUEncoded data
P_UI_INFO_MIME	5	The information to be sent to the end-user consists of MIME encoded data
P_UI_INFO_WAVE	6	The information to be sent to the end-user is .wav waveform data
P_UI_INFO_AU	7	The information to be sent to the end-user is .au audio data
P_UI_INFO_VXML	8	The information to be sent to the end-user is controlled by this VXML
P_UI_INFO_SYNTHESIS	9	The information to be sent to an end-user is synthesized from text.

11.28 TpUISynthesisInfoData

Defines the Sequence of Data Elements that specify the information to use in generating the desired effects on the generated voice. The speech synthesis parameters or processing tags will be interpreted as hints and may be ignored by the speech synthesis engine. Note that the language is specified on the sendInfoReq() and sendInfoAndCollectReq() method calls.

The TextData field may contain the following tags to affect the processing of the text. The <break> tag specifies a timing pause. The <emp> tag specifies an emphasis on a word or phrase.

```
<break size="milliseconds">      // specifies a timing pause in the
                                 // middle of the text in milliseconds
<emp>A word</emp>                // apply emphasis to a word or words
```

Structure Element Name	Structure Element Type	Structure Element Description
SpeakerGender	TpUISynthesisGender	Defines the gender of the speaker.
SpeakerAge	TpUISynthesisAge	Defines the age of the speaker.
SpeakerRate	TpUISynthesisRate	Defines the rate of the speaker.
SpeakerRange	TpUISynthesisRange	Defines the range of the speaker
TextData	TpString	Defines the text to synthesize into speech.
WordOverrideList	TpUIWordOverrideList	Defines the pronunciation overrides

11.29 TpUISynthesisGender

Defines the UI reports if a response was requested.

Name	Value	Description
P_UI_GENDER_MALE	0	Male
P_UI_GENDER_FEMALE	1	Female

11.30 TpUISynthesisAge

Defines the UI reports if a response was requested.

Name	Value	Description
P_UI_AGE_CHILD	0	child voice
P_UI_AGE_YOUNG_ADULT	1	Young adults voice
P_UI_AGE_ADULT	2	Adult voice
P_UI_AGE_OLDER_ADULT	3	Older adult voice

11.31 TpUISynthesisRate

Defines the rate of the speech.

Name	Value	Description
P_UI_RATE_SLOW	0	Slow speech rate
P_UI_RATE_AVERAGE	1	Average speech rate
P_UI_RATE_FAST	2	Fast speech rate

11.32 TpUISynthesisRange

Defines the range or liveliness of the speech.

Name	Value	Description
P_UI_RANGE_CALMER	0	Very Calm or monotone speech
P_UI_RANGE_CALM	1	Moderately calm speech
P_UI_RANGE_AVERAGE	2	Average speech
P_UI_RANGE_EXCITED	3	Moderately excited or lively speech
P_UI_RANGE_MORE_EXCITED	4	Excited or lively speech

11.33 TpUIWordOverrideList

Defines a `Numbered Set of Data Elements` of TpUIWordOverride.

11.34 TpUIWordOverride

Defines the `Sequence of Data Elements` that specify the information to use in overriding the default pronunciation of a word.

Structure Element Name	Structure Element Type	Structure Element Description
Spelling	TpString	Defines the spelling of the word override.
PronounceType	TpUIPronounceType	Defines the type of pronunciation syntax.
PronounceAs	TpString	Defines how the spelling field should be pronounced.

11.35 TpUIPronounceType

Defines the pronunciation type.

The International Phonetic Alphabet (IPA) representation can be used to specify pronunciations. For more information see:

http://www2.arts.gla.ac.uk/IPA/ipachart.html
http://charts.unicode.org/Unicode.charts/normal/U0250.html

Also, simple sound-alike replacements can be used, such as "I triple E" for IEEE.

Name	Value	Description
P_UI_PRONOUNCE_IPA	0	The IPA pronunciation type
P_UI_PRONOUNCE_SOUNDSLIKE	1	The simple sounds like replacement type

11.36 TpUICollectMode

Defines the type of collection.

Name	Value	Description
P_UI_COLLECT_MODE_DTMF	0	Collect DTMF digits only
P_UI_COLLECT_MODE_VOICE	1	Collect Voice recognized data
P_UI_COLLECT_MODE_DMTFANDVOICE	2	Collect both DTMF digits and voice recognized data

11.37 TpUIRecognitionCriteria

Defines the `Sequence of Data Elements` that specify the additional properties for the collection of information in the form of voice recognition according to the specified grammar.

Structure Element Name	Structure Element Type
SpeakerID	TpUIRecognitionSpeakerID
Properties	TpUIRecogitionPropertyList
Grammar	TpUIRecognitionGrammar

The structure elements specify the following criteria:

`SpeakerID:` Defines the user identifier associating a user with a speech profile known to the recognition engine, which provides a hint for better quality.

`Properties:` Defines the properties list for additional information to the speech recognition engine.

`Grammar:` Defines the syntax of the language to be recognized.

11.38 TpUIRecognitionSpeakerID

Defines a user identifier string that identifies the speaker and is a hint to whose voice is to be recognized.

11.39 TpUIRecognitionPropertyList

Defines a set of TpUIRecognitionProperty objects.

11.40 TpUIRecognitionProperty

Defines the `Sequence of Data Elements` that specify the additional properties for the recognition engine. The TpUIRecognitionProperty is a hint to the recognition engine on how it should interpret the input.

Structure Element Name	Structure Element Type
`PropertyName`	`TpString`
`PropertyValue`	`TpString`

The structure elements specify the following criteria:

`PropertyName:` Defines the name of the property.

`PropertyValue:` Defines the value of the property.

The defined properties are:

P_RECOGNITION_PROPERTY_CONFIDENCE_LEVEL - The speech recognition confidence level, a float value in the range of 0.0 to 1.0. Results are rejected when the recognitions engine's confidence in its interpretation is below this threshold. A value of 0.0 means minimum confidence is needed for a recognition, and a value of 1.0 requires maximum confidence. The default value is 0.5.

P_RECOGNITION_PROPERTY_SENSITIVITY - Set the sensitivity level. A value of 1.0 means that it is highly sensitive to quiet input. A value of 0.0 means it is least sensitive to noise. The default value is 0.5.

P_RECOGNITION_PROPERTY_SPEEDVSACCURACY - A hint specifying the desired balance between speed vs. accuracy. A value of 0.0 means fastest recognition. A value of 1.0 means best accuracy. The default is value 0.5.

P_RECOGNITION_PROPERTY_COMPLETE_TIMEOUT - The speech timeout value to use when an active grammar is matched.

11.41 TpUIRecognitionGrammar

Defines a string that consists of an inline grammar that specifies the syntax of the speech to be recognized. The format of this string is a subset of the Voice XML 1.0 grammar element tag. The <grammar> element tag 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 of Voice XML version 1.0 and the Java Speech Grammar Format, see:

http://java.sun.com/products/java-media/speech/forDevelopers/JSGF/index.html