

フロントエンドの仕様について

入力モダリティ

<input>では、属性 type に入力モダリティを指定し、属性 event に発生するイベントを指定する。

入力モード (mode) の指定は、<input>の子要素<param>で行う。

type	event	mode
dtmf	push	digit,character
speech	recognize	command, data, speaker,emotion など
pointing	l_click, r_click など	
kyeboard	press	

□ DTMF

入力モダリティ	type属性	操 作	event属性	match属性	return属性
DTMF	"dtmf"	プッシュ	"push"	認識した生成規則へのパス	1:取得した言葉とマッチした生成規則同名の変数(複数指定が可能)

認識モードについて

認識モードには、

- digit
- character

などが考えられる。

例

```
<input type="dtmf" event="push" target="grammar.gxml"
      match="/grammar/goods_select" >
  <param name="mode">digit </param>
</input>
```

□ 音声

入力モダリティ	type属性	操 作	event属性	match属性	return属性
音声	"speech"	認識完了	"recognize"	認識した生成規則へのパス	1:取得した語彙とマッチした生成規則同名の変数(複数指定が可能)

認識モードについて

認識モードには、

- command
- data
- speaker
- emotion

などが考えられる。

このような認識モードの選択・切り替えは、<input>の子要素<param>で指定する。

例

```
<input type="speech" event="recognize" target="grammar.xml"
      match="/grammar/goods" return="goods_name" >
  <param name="mode"> data </param>
</input>
```

□ キーボード

入力モダリティ	type属性	操 作	event属性	match属性	return属性
キーボード	"keyboard"	キー入力	"press"	操作対象のコンテンツのルートノード	1:入力された文字

□ ボインティング

入力モダリティ	type属性	操 作	event属性	match属性	return属性
ポインティング	"pointing"	クリック	左"l.click" 右"r.click"	入力の対象となる オブジェクトのパス	1:イベントが発生したオブジェクト 2:イベントが発生した位置のx座標 3:イベントが発生した位置のy座標
		ダブルクリック	左"ld.click" 右"rd.click"		
		マウスオーバー	"over"		
		マウスアウト	"out"		
		ドラッグ&ドロップ	左"l.drag" 右"r.drag"	ドラッグを開始する オブジェクトのパス	1:ドラッグを開始したオブジェクト 2:ドラッグを終了したオブジェクト 3:ドラッグを開始した位置のx座標 4:ドラッグを開始した位置のy座標 5:ドラッグを終了した位置のx座標 6:ドラッグを終了した位置のy座標

出力モダリティ

type	event	mode
window	open,close など	
speech	play,stop など	TTS,rec_speech など
video	play,stop など	
audio	play,stop など	
agent	move,speak など	

□ ウィンドウ

出力モダリティ	type属性	動作	event属性	<param>	
				name属性	値
ウィンドウ	window	ウィンドウを開く	"open"	window_name	ウィンドウにつける名前
				window_type	ウィンドウの表示形式[省略可] ノーマル:"normal"(デフォルト) モーダルウィンドウ:"modal"
				uri	表示するページのURI[省略可能]
				position	ウィンドウの初期位置[省略可] "move" eventを参照
				width	ウィンドウの初期サイズ[省略可] "resize" eventを参照
				height	
		コンテンツを読み込む	"navigate"	window_name	操作対象のウィンドウ名
				uri	表示するページのURI
		ウィンドウを閉じる	"close"	window_name	操作対象のウィンドウ名
		ウィンドウを最大化	"maximize"		
		ウィンドウを最小化	"minimize"		
		元のサイズに戻す	"restore"		
		ウィンドウのサイズを変更する	"resize"	window_name	操作対象のウィンドウ
				width	ウィンドウの幅
				height	ウィンドウの高さ
		ウィンドウを移動する	"move"	window_name	操作対象のウィンドウ
				position	ウィンドウの移動先 中央:"center", 左上:"left-to" 右上:"right-top", 左下:"left-bottom" 右下:"right-bottom"

□ 音声

・TTS モード

出力モダリティ	type属性	動 作	event属性	＜param＞	
				name属性	値
音声	"speech"	音声出力	"play"	id	操作対象のID
				mode	TTS
				speech_text	読み上げテキスト
				eibun_flag	英語読みをするかスペルのまま読むか
				num_flag	数字を棒読みするか、桁読みをするか
				speed	読み上げ速度[省略可]
				volume	音量 [省略可]
				speaker	音声の種類(男性、女性など)
				emotion	感情
		停止	"stop"	id	操作対象のID
		一時停止	"pause"		

・rec_speech モード

出力モダリティ	type属性	動 作	event属性	＜param＞	
				name属性	値
音声	"speech"	音声出力	"play"	id	捜査対象のID
				mode	rec_speech
				uri	再生するファイルのURI
				volume	音量 [省略可]
		停止	"stop"	id	操作対象のID
		一時停止	"pause"		

□ 動画

出力モダリティ	type属性	動作	event属性	<param>	
				name属性	値
動画	video	ファイルを開く	“open_file”	id	操作対象のID
				uri	表示するファイルのURI
		再生の開始・再開	“play”	id index	操作対象のID,index
		再生の停止	“stop”		
		再生の一時停止	“pause”		

□ オーディオ

出力モダリティ	type属性	動作	event属性	<param>	
				name属性	値
オーディオ	audio	ファイルを開く	“open_file”	id	操作対象のID
				uri	表示するファイルのURI
		再生の開始・再開	“play”	id index	操作対象のID,index
		再生の停止	“stop”		
		再生の一時停止	“pause”		

□ 擬人化エージェント

出力モダリティ	type属性	動 作	event属性	<param>	
				name属性	値
擬人化エージェント	“agent”	エージェント生成	“create”	agent_name	エージェントにつける名前
				agent_file	表示するキャラクターファイル
		エージェント表示	“show”	agent_name	操作対象のエージェント名
		エージェント非表示	“hide”		
		エージェント削除	“delete”		
		エージェント移動	“move”	agent_name	操作対象のエージェント名
				① ax	移動先のx座標(ウインドウ座標)
				ay	移動先のy座標(ウインドウ座標)
				object	移動先のオブジェクト(XMLのパス)
				② position	オブジェクト上のどの位置に移動するか 中央:“center”, 左:“left”, 右:“right”
		アクション出力	“action”	agent_name	操作対象のエージェント名
				action_name	アクション名
		バルーン出力	“balloon”	agent_name	操作対象のエージェント名
				balloon_text	表示テキスト
		スピーチ出力	“speak”	agent_name	操作対象のエージェント名
				speech_text	読み上げテキスト
				agent_name	操作対象のエージェント名
		キャラクタのVoice変更	“change_voice”	voice	LaLaVoiceで指定できるキャラクター お兄さん:“brother”, お姉さん:“sister” お坊ちゃん:“boy”, お嬢ちゃん:“girl” おじいさん:“g.father”, おばあさん:“g.mother” ロボットA:“robotA”, ロボットB:“robotB” ロボットC:“robotC”