

condition is returned the QA control is activated. The use of an activation mechanism allows increased flexibility and improved dialog flow in the client side markup page produced. As indicated in Appendix B many of the controls and objects include an activation mechanism.

[0096] Command Control

[0097] Command controls **310B** are user utterances common in voice-only dialogs which typically have little semantic import in terms of the question asked, but rather seek assistance or effect navigation, e.g. help, cancel, repeat, etc. The Command control **310B** within a QA control **306** can be used to specify not only the grammar and associated processing on recognition (rather like an answer control **310A** without binding of the result to an input field), but also a ‘scope’ of context and a type. This allows for the authoring of both global and context-sensitive behavior on the client side markup.

[0098] As appreciated by those skilled in the art from the foregoing description, controls **306** can be organized in a tree structure similar to that used in visual controls **302**. Since each of the controls **306** are also associated with selected visual controls **302**, the organization of the controls **306** can be related to the structure of the controls **302**.

[0099] The QA controls **302** may be used to speech-enable both atomic controls (textbox, label, etc.) and container controls (form, panel, etc.) This provides a way of scoping behaviour and of obtaining modularity of subdialog controls. For example, the scope will allow the user of the client device to navigate to other portions of the client side markup page without completing a dialog.

[0100] In one embodiment, “Scope” is determined as a node of the primary controls tree. The following is an example “help” command, scoped at the level of the “Pn1” container control, which contains two textboxes.

```

<asp:panel id="Pn1" . . . >
  <asp:textbox id="tb1" . . . />
  <asp:textbox id="tb2" . . . />
</asp:panel>
<Speech:QA ... >
  <Command
    id="HelpCmd1"
    scope="Pn1"
    type="help"
    onClientReco="GlobalGiveHelp ( )" >
    <Grammar src="grammars/help.gram" />
  </Command>
</Speech:QA>
<script>
  function GlobalGiveHelp ( ) {
    . . .
  }
</script>

```

[0101] As specified, the “help” grammar will be active in every QA control relating to “Pn1” and its contents. The GlobalGiveHelp subroutine will execute every time “help” is recognized. To override this and achieve context-sensitive behavior, the same typed command can be scoped to the required level of context:

```

<Speech:QA . . . >
  <Command
    id="HelpCmd2"
    scope="Tb2"
    type="help"
    onClientReco="SpecialGiveHelp ( )" >
    <Grammar src="grammars/help.gram" />
  </Command>
</Speech:QA>
<script>
  function SpecialGiveHelp ( ) {
    . . .
  }
</script>

```

[0102] Confirmation Control

[0103] The QA control **320** can also include a method for simplifying the authoring of common confirmation subdialogs. The following QA control exemplifies a typical subdialog which asks and then confirms a value:

```

<Speech:QA
  id="qaDepCity"
  controlsToSpeechEnable="txtDepCity"
  runat="server" >
  <!-- asking for a value -->
  <Question id="AskDepCity"
    type="ask"
    Answers="AnsDepCity" >
    <prompt> Which city? </prompt>
  </Question>
  <Answer id="AnsDepCity"
    confirmThreshold="60" >
    <grammar src="grammars/depCity.gram"
  />
  </Answer>
  <!-- confirming the value -->
  <Confirm id="ConfirmDepCity"
    Answers="AnsConfDepCity" >
    <prompt>
      Did you say <value
      targetElement="txtDepCity/Text">?
    </prompt>
  </Confirm>
  <Answer id="AnsConfDepCity" >
    <grammar
      src="grammars/YesNoDepCity.gram" />
  </Answer>
</Speech:QA>

```

[0104] In this example, a user response to ‘which city?’ which matches the AnsDepCity grammar but whose confidence level does not exceed the confirmThreshold value will trigger the confirm control **308**. More flexible methods of confirmation available to the author include mechanisms using multiple question controls and multiple answer controls.

[0105] In a further embodiment, additional input controls related to the confirmation control include an accept control, a deny control and a correct control. Each of these controls could be activated (in a manner similar to the other controls) by the corresponding confirmation control and include grammars to accept, deny or correct results, respectively. For instance, users are likely to deny by saying “no”, to accept by saying “yes” or “yes+current value” (e.g., “Do you want