

In this way the user is able to ask for help at any time, as he does in turn 13. Command controls may also be more sensitively enabled by a mechanism that scopes their activation according to which part of the primary control structure is being talked about.

[0151] Referring back to the algorithm, in one exemplary embodiment, the client-side script RunSpeech examines the values inside each of the primary controls and an attribute of the QA control, and any selection test of the QA controls on the current page, and selects a single QA control for execution. For example, within the selected QA control, a single question and its corresponding prompt are selected for output, and then a grammar is activated related to typical answers to the corresponding question. Additional grammars may also be activated, in parallel, allowing other commands (or other answers), which are indicated as being allowable. Assuming recognition has been made and any further processing on the input data is complete, the client-side script RunSpeech will begin again to ascertain which QA control should be executed next. An exemplary implementation and algorithm of RunSpeech is provided in Appendix B.

[0152] It should be noted that the use of the controls and the RunSpeech algorithm or module is not limited to the client/server application described above, but rather can be adapted for use with other application abstractions. For instance, an application such as VoiceXML, which runs only on the client device 30, could conceivably include further elements or controls such as question and answer provided above as part of the VoiceXML browser and operating in the same manner. In this case the mechanisms of the RunSpeech algorithm described above could be executed by default by the browser without the necessity for extra script. Similarly, other platforms such as finite state machines can be adapted to include the controls and RunSpeech algorithm or module herein described.

[0153] Synchronization

[0154] As noted above, the companion controls 306 are associated with the primary controls 302 (the existing controls on the page). As such the companion controls 306 can re-use the business logic and presentation capabilities of the primary controls 302. This is done in two ways: storing values in the primary controls 302 and notifying the primary controls of the changes 302.

[0155] The companion controls 306 synchronize or associate their values with the primary controls 302 via the mechanism called binding. Binding puts values retrieved from recognizer into the primary controls 302, for example putting text into a textbox, herein exemplified with the answer control. Since primary controls 302 are responsible for visual presentation, this provides visual feedback to the users in multimodal scenarios.

[0156] The companion controls 306 also offer a mechanism to notify the primary controls 302 that they have received an input via the recognizer. This allows the primary controls 302 to take actions, such as invoking the business logic. (Since the notification amounts to a commitment of the companion controls 306 to the values which they write into the primary controls 302, the implementation provides a mechanism to control this notification with a fine degree of control. This control is provided by the RejectThreshold and ConfirmThreshold properties on the answer control, which specify numerical acoustic confidence values below which the system should respectively reject or attempt to confirm a value.)

[0157] Although the present invention has been described with reference to preferred embodiments, workers skilled in the art will recognize that changes may be made in form and detail without departing from the spirit and scope of the invention.