the question, or are they correcting the value, or are they asking for help? By using the role of the user input in the dialogue the system can know when to implicitly accept a value.

[0130] In summary, a dialog is created due to the role of the control in the dialog and the relationship with other controls, wherein the algorithm executes the controls and thus manages the dialog. Each control contains information based on its type which is used by the execution algorithm to select (i.e. make active for execution) a given control according to whether or not it serves a useful purpose at that point in the dialog on the client. For example, confirmation controls are only active when there is a value to confirm and the system does not have sufficient confidence in that value to proceed. In a further implementation, most of these built-in pieces of information can be overridden or otherwise adapted by application developers.

[0131] The following table summarizes the controls, their corresponding role in the dialog and the relationship with other controls.

Control	Role in dialogue	Relationship with other controls
Statement	output: present information to users	(none)
Question	output: ask question	selects expected Answers as a response
Confirmation	output: confirm a value obtained from the user	selects potential input controls as a response, typically Accept, Deny, Correct
Answer	input: provide an answer to a question	selected by Question/Confirmation
Command	input: seek to repair a dialog, or change the topic of conversation	scoped to other controls
Accept	input: confirm a value in response to a confirmation	selected by a confirmation
Deny	input: deny a value in response to a confirmation	selected by a confirmation
Correct	input: correct a value in response to a confirmation	selected by a confirmation
QA		(wrapper: contains all the controls related to a specific topic)

[0132] The use of these controls may be explained with an illustration of a simple human/computer dialog. In the dialog below, each dialog turn on the part of the System or the User is characterized according to the control (indicated in parentheses) which reflects its purpose in the dialog.

[0133] 1. System (Statement): "Welcome to the travel booking service".

[0134] 2. System (Question): "Where would you like to go?"

[0135] 3. User (Answer): "San Francisco."

[0136] 4. System (Confirmation): "Did you say Seattle?"

[0137] 5. User (Deny): "No."

[0138] 6. System (Question): "Where would you like to go?"

[0139] 7. User (Answer): "San Francisco."

[0140] 8. System (Confirmation): "Did you say Seattle?"

[0141] 9. User (Correct): "I said San Francisco."

[0142] 10. System (Confirmation): "Did you say San Francisco?"

[0143] 11. User (Correct): "Yes."

[0144] 12. System (Question): "When would you like to leave?"

[0145] 13. User (Command): "Help."

[0146] Turn 1 is a statement on the part of the System. Since a statement control activates no answer controls in response, the system does not expect input. The system goes on to activate a question control at turn 2. This in turn activates a set of possible answer controls, including one which holds a grammar containing the cities available through the service, including "San Francisco", "Seattle", etc., which permits the user to provide such a city in turn 3.

[0147] The user's turn 3 is misrecognized by the system. Although the system believes it has a value from an answer control for the city, its confidence in that value is low (rightly so, since it has recognized incorrectly). This low confidence value in a just-received answer control is sufficient information for RunSpeech to trigger a confirmation control on the part of the system, as generated at turn 4. The confirmation control in turn activates a deny control, a correct control and an accept control and makes their respective grammars available to recognize the user's next turn. User turns 5, 9 and 11 illustrate example responses for these controls. Turn 5 of the user simply denies the value "no". This has the effect of removing the value from the system, so the next action of RunSpeech is to ask the question again to re-obtain the value (turn 6).

[0148] Turns 7 and 8 return us to a confirmation control as with 3 and 4.

[0149] User turn 9 is a correct control, which has again been activated as a possible response to the confirmation control. A correct control not only denies the value undergoing confirmation, it also provides a new value. So user turn 9 is recognized by the system as a correct control with a new value which, correctly this time, is recognized as "San Francisco".

[0150] The system's confidence in the new value is low, however, and yet another confirmation control is generated at turn 10. This in turn activates accept, deny and correct controls in response, and user turn 11 ("Yes" matches an accept control grammar. The recognition of the accept control has the effect of 'grounding' the system's belief in the value which it is trying to obtain, and so RunSpeech is now able to select other empty values to obtain. In turn 12, a new question control is output which asks for a date value. The user's response this time (turn 13) is a command: "help". Command controls are typically activated in global fashion, that is, independently of the different question controls and confirmation controls on the part of the system.