Jon Garibay

4/8/2011

Intro

The PAC calculator was an experiment into the implementation of the Presentation-abstraction-control architecture. This creates highly modular code that can easily have pieces inserted and removed without the need for recoding. The calculator accepts three types of input. These inputs are voice, buttons, and keyboard. The calculator also has two forms of output, a Panel GUI and Voice output.

Design

Each individual model implements the "Agent" interface. This abstraction allows us to plug modules into the program without changing variable types. The Agent contains four methods, getAgent(), setAgent(Agent[] a1), SendMessage(String s) and RcvMessage(String s). The setAgent (Agent[] a1) method allows the programmer to add agents that will need to be messaged when events happen. All Agents in the array a1 will be sent all messages through the SendMessage(String s) method. The SendMessage(String s) method simply calls the RcvMessage(String s) method of each Agent. It is up to the individual Agents to process or ignore message according to their RcvMessage(String s) implementation. An example of this from the implemented architecture (fig. A) is CalculatorAgent would be initialized and then setAgent(a1) would be called with a1 being the array of size 2 containing PanelAgent and VoiceOutAgent. Then any SendMessage(String s) called within the CalculatorAgent would send the message to both the PanelAgent and VoiceOutAgent through their respective RcvMessage(String s) methods.

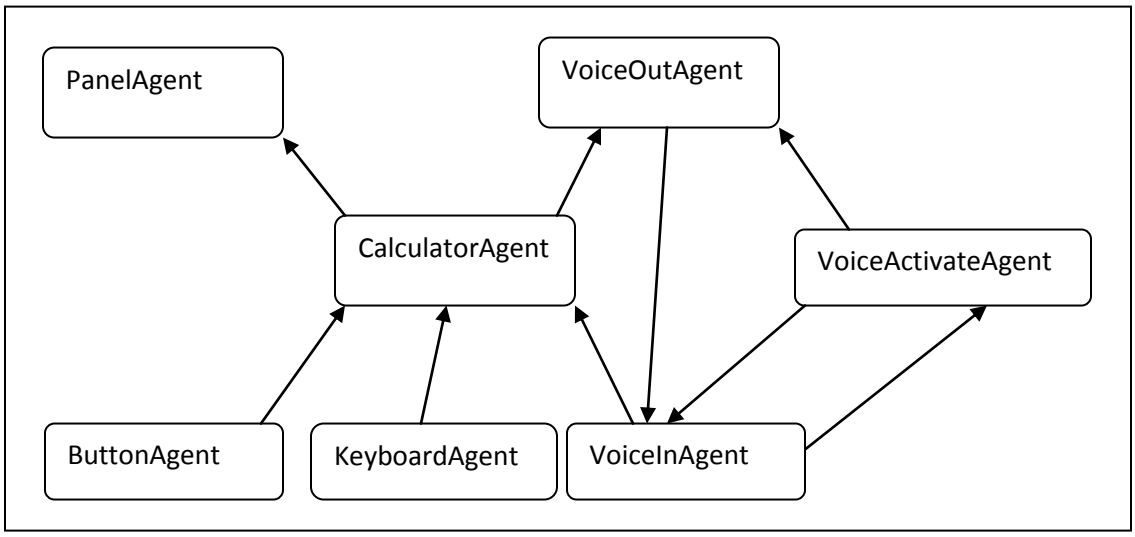

fig. A

It is also expected that any mid level agent (i.e. CalculatorAgent) sends along all message sent to it. This ensure seamless operation if the CalculatorAgent was removed. While a strictly linear implementation was desired, to ensure proper operation of the voice input/output, additional connection needed to be implement. The VoiceOutAgent needs to inform the VoiceInAgent when it is talking in order to stop outputted speech from being recognized as voice input. The VoiceInAgent must also notify the VoiceActivateAgent as the voice input/output can be activated through a voice command.

Modules can easily be added to the calculator at any level through the Agent interface. Since Agents only process expected messages and pass the rest along, any extra messages a new input Agent would send would not affect the operation of the calculator. Accepted messages can be found in the documentation for each agent. Output Agents can also use the documentation for lower level agents to process any messages it may receive from them. Once the module is designed correctly, the programmer will only need to call setAgent(a1) accordingly.

Error Handling

The calculator has simplistic error handling strategy. Errors will only occur when the equation is being calculated and the solution cannot be found. The program will simply output the answer as "Error" and will output the exception to the console. This accurately conveys that the equation was unsolvable allowing the user to fix the mistake and resubmit the equation.

Additional JavaDoc Documentation