

Adding Voice Control To A Seminar Room

Stuart Vaile

Seng Loke

Chris Ling



Purpose

The purpose of this project is to investigate the viability of controlling a seminar room using simple voice commands.

The goals of this project are to:

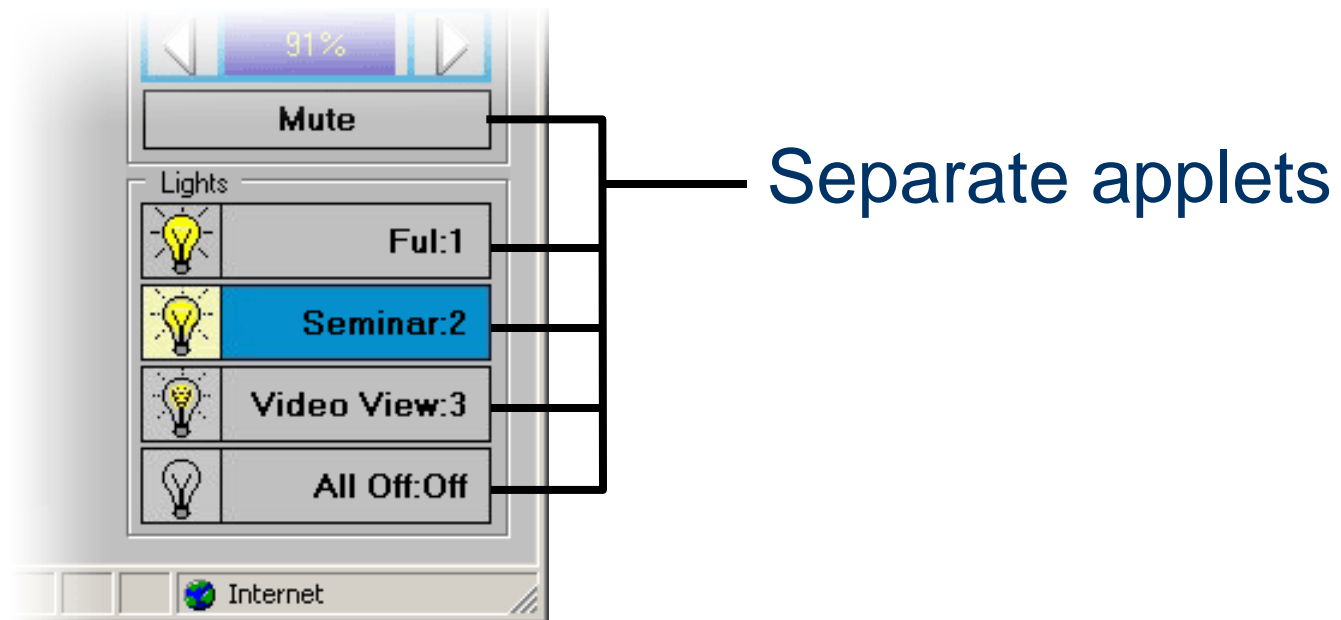
- Be able to control the devices in the seminar room using simple speech commands.
- Be able to control Microsoft PowerPoint slide show using simple speech commands.

Overview

- Existing System
- The Addition Of Speech Control
- The Implementation and Uses
- Limitations
- Conclusions

The Existing System

The seminar room contains a web interface for controlling the devices. The devices are controlled by applets embedded in the web pages.



The Addition of Speech Control

The speech control program uses the existing web interface to control the devices.

Navigation and activation of a device is performed by:

- simulating user selection on the applets or anchors by programmatically moving the mouse and simulating a mouse click; or
- simulating a selection on an anchor.

The speech control users OLE Automation to access and control the internals of Internet Explorer and PowerPoint.

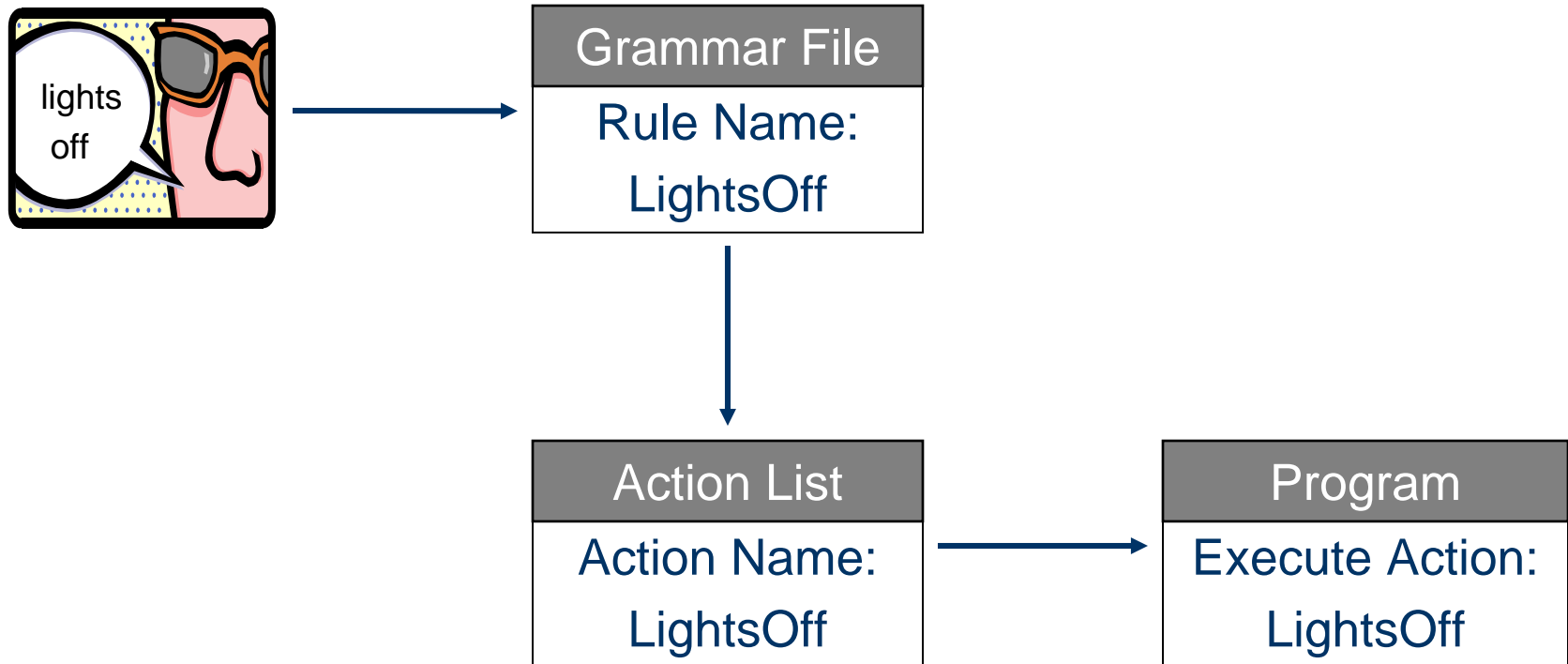
The Speech Control Program

The program associates a spoken command with a named action.

The program defines two different types of actions:

- Simple Action: Anchor, Applet, Absolute Point, General, PowerPoint; and
- Composite Action: A sequence of simple actions.

From Speech To Action



Simple Actions

Have an action name associated with a particular function.

The functionality for PowerPoint and General actions have been hard coded into the program and cannot be added without recompiling.

All *Applet*, *Anchor*, and *Absolute Point* actions can be created at runtime. A phrase rule must be added to a grammar before the action can be used.

Examples of Simple Actions

Spoken commands:

- “show previous slide”
- “show next slide”
- “goto slide four”

Action Names:

- *LightsFull-Abs*
- *LightsOff-Abs*

NB: The actual “lights full” and “lights off” spoken phrases use composite actions.

Composite Actions

Composite Actions perform a sequence of Simple Actions.

The user can define a delay time between each simple action.

The user can also define an argument to pass to the simple action. In the case of providing an action with a spoken number.

- Eg. “move forward five”

Examples of Composite Actions

Spoken commands:

- “lights full”, “lights off”, “lights Seminar”, “lights Video”
- “question time”
- “presentation my seminar two”

Composite Actions can be defined at runtime after creation of a phrase rule in the grammar.

Creating A Simple Action

To add a speech command the user needs to edit the grammar file and add a rule.

```
<RULE NAME="lightsFull12" TOPLEVEL="ACTIVE">  
    <P>turn lights on</P>  
</RULE>
```

For this example it has been done already.

Creating A Simple Action (Cont.)

The user can type in the values or for the Anchor, Applet, and Absolute Point they can record a macro. The *Action Name* must be the same as the corresponding rule name in the grammar.

Recording a macro involves:

1. Selecting the type of action (*Anchor, Applet* or *Absolute Point*)
2. Selecting *Record Macro*
3. Moving the mouse over the *Anchor, Applet*, or point of interest
4. Holding the left mouse button down

Creating A Composite Action

The user first needs to add a phrase rule in the grammar.

```
<RULE NAME="JumpAround" TOPLEVEL="ACTIVE">
    <P>jump to <RULEREF PROPNAME="PLACE1" NAME="Numbers" />
    then back to <RULEREF PROPNAME="PLACE2"
    NAME="Numbers" /></P>
</RULE>
```

The `<RULEREF...>` is a reference to another rule. In this case it refers to a sequence of spoken numbers.

Creating A Composite Action (Cont.)

Composite actions can contain any simple actions.

The *Action Name* must be the same as the corresponding rule name in the grammar.

To create the sequence:

1. Choose *Sequence* as the type
2. Add actions in the order that they should be performed
3. Change the delay time and argument if required

Restoring The Active Window

A composite action can restore the window that was active before Internet Explorer was brought to focus.

This is done by placing the *RestoreActive* action as the last action in the sequence.

The spoken commands such as “lights full”, “lights off”, and “question time” use this feature.

Limitations

Anchor & Applet detection:

For the anchor and applet detection and selection to work properly the anchors and applets need *ID* tags. We were unable to alter the vendor provided web interface.

The program tries to add unique *IDs* to any anchors and applets that do not have them after the page has been loaded.

This sometimes fails due to Internet Explorer's inconsistent throwing of events.

The solution is to define the action as an Absolute Point.

Conclusions

The system itself functions reasonably well, although sometimes a spoken command is interpreted incorrectly or continuous speech falsely triggers a command.

It's not totally intuitive. The user needs to know the exact commands that will induce a response. It could be reasonably intuitive with well chosen commands or many command options.

Future Work

The main drawback to this system is that it is not intelligent. The user needs to explicitly command the system.

A system that could alter the state of the room and the PowerPoint slide show on the perceived state of the presenter's dialog would be less obtrusive than the current system.



Questions?

