Simulating Speech Interfaces by Constructing a GUI for a Light Control System

Dujuan Williams, Ana Ramirez Chang, John Canny

Berkeley Institute of Design
Department of Electrical Engineering and Computer Science
University of California, Berkeley
wil9844@setonhill.edu

Abstract
The purpose of my project is to build a graphical user interface to control a light system in a wizard of Oz study that will aid in the development of a new speech interface for the light control system in a lab. The speech interface will serve the role of a test application for conducting research in speech interfaces. The development of a speech interface system requires paying very close attention to the unique qualities of the interaction between humans and machines. Designing a speech interface system is hard because designers must model as closely as possible the dialogue between humans. But the fact that computers are not humans suggest that we cannot use human to human data as a viable source of design data for the creation of a new speech interface system. Instead, a wizard of Oz study is used, where a human simulates an implemented system, allowing designers to study the language that is used in a unique dialogue situation between users and a computer. The goal of the wizard of Oz study is to collect data to inform the design of a speech interface for the lighting in the lab. The graphical user interface (GUI) for the light system will allow the wizard (the human simulating the system) in the Wizard of Oz study to respond to the user's requests more quickly, allowing him to more closely simulate an implemented speech interface to the lighting in the lab. This will allow us, to collect more realistic data for the future speech interface.

Introduction
Speech interfaces and applications present new ways of interaction for end-users to fully express human-to-human dialogue in terms of semantic expressions. [5] A major benefit of incorporating speech interfaces within applications is that speech is thought of as a natural process; people find speech as an easy task, a skill learned from the early stages of our lifetime. [1] Applying speech interfaces to applications is seen as instantiating a conversation between the end-user and the computer system. The Development of a speech interface system requires paying very close attention to the unique qualities of the interacting between humans and machines. Designing a speech interface system is hard because designers must model as closely as possible the dialogue between humans.

The success of a speech interface system is determined on the basis of whether or not there is a clear benefit to using speech within a given system. Speech is seen as a natural process, it doesn't take much effort in constructing a sentence and expressing them in words, therefore end-users' have high expectations when they are faced with a system that is based on speech. This suggests that speech is best used when the need is clear - for example, when the user's hands, eyes are busy, environmental constraints are present or when speech enables something that cannot otherwise be done. [1]. Speech applications are suited best for environments where the end-user is motivated to participate with the system.

Use When...	Avoid When...
No keyboard is available (e.g., over the telephone, at a kiosk, or on a portable device). Task requires the user's hands to be occupied so they cannot use a keyboard or mouse (e.g., maintenance and repair, graphics editing). Commands are embedded in a deep menu structure. Users are unable to type or are not comfortable with typing. Users have a physical disability (e.g., limited use of hands).	Task requires users to talk to other people while using the application. Users work in a very noisy environment. Task can be accomplished more easily using a mouse and keyboard.

Figure 1: Table Modified, Showing When Speech Input is Appropriate [1]

For example, the GUI for my Light Control System allows the wizard, in a Wizard of Oz study to individually access, manipulate and/or change the light settings for the lights that are in the wizards' view. An End-user would no longer have to manually get up and turn on/off at their discretion; they would simply be able to use the advantages and ease of speech to turn on/off any and all lights that they wanted to control.

The Graphical User Interface tool that I created will allow the wizard, in a Wizard of Oz Study to control the light system within a lab for the study. Light Control System and it will help in performing research necessary for modeling and designing the future speech interface system. The GUI will provide data, in terms of the corpus, domain vocabulary, user utterances, and semantic grammar representation to design the future speech interface system for controlling the lights in the lab. The GUI also allows me to control the light system within a lab for a Wizard of Oz Study. Data gathered from the use of the GUI for the light system in the lab will provide important clues and results as to what necessary input would have to be generated by the future speech interface system in order to model as closely the utterances that are sent from the end-user.

Wizard of Oz Study

The wizard in this study will be able to capture all descriptors that are necessary for the general functionality of the light system and then use this information to simulate an implemented speech interface that is as close to real-time requests as possible. The GUI will provide an accurate research tool for designing realistic dialogue between humans and speech interface systems. Involving users in the design process throughout the lifecycle of a speech interface system is crucial. A natural and effective interface can only be achieved by understanding the how, where, and why users would interact with an application. Natural Dialogue Studies are a necessity in the design process for the future speech interface. At the early stages of our speech interface design, most normally during the preliminary application design, Natural Dialogue Studies will be used as an effective method in collecting domain-specific vocabulary for speech interface systems. [4] Natural Dialogue Studies will help in defining the application functionality for the future speech interface. The fact that computer is not a human suggests that we cannot use human-to-human data as a viable source of design data for the creation of a new speech interface system. Instead, a wizard of Oz study is used, where a human simulates an implemented system using software tools, allowing designers to study the language that is used in a unique dialogue situation between uses and a computer. The goal of the wizard of Oz study is to collect data to inform the design of a speech interface for the lighting in the lab. The Wizard of Oz study would help us to study how end-users interact with the speech interface system and register the types of speech relationships that are involved in such a system. [4] Wizard of Oz Studies helps in the testing and refinement for the speech interface; the study will provide invaluable discoveries and possible usability issues. The graphical user interface for the light system will allow the wizard (the human simulating the system) in the Wizard of Oz study to respond to the user's requests more quickly, allowing him to more closely simulate an implemented speech interface to the lighting in the lab. We must design some type of reasonable approach to handle errors both with regard to the light in the Light System Control that a end-user may wish to manipulate and also the Natural Command Languages that would translate the user's request to a semantic expression that would intern send commands to the GUI. The Wizard of Oz study will provide accurate data that can help future speech interface developers to construct procedures, methods, and algorithms to correct transience, asymmetry, and synthesis recognition quality in speech system interfaces.

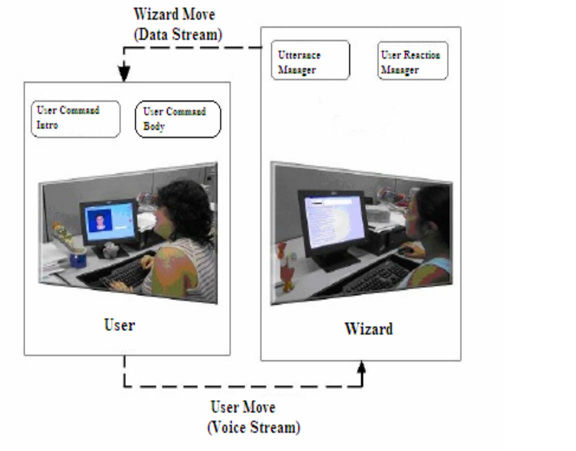

Figure 2: (Modified) Wizard of Oz Study for GUI [2]

An effective speech application is one that uses speech to enhance end-user's performance of a task. "Designing an application with speech in mind from the outset is a key success factor." [1] Basing the dialog design on a natural dialog study ensures that the input grammar and the vocabulary-domain will match the phrasing precisely as used by people when speaking in the domain of the application. Natural Dialogue Studies will aid in the overall design performance and integrity for the future speech interface by establishing concrete and well-defined corpus. The GUI will be used to allow the wizard in the Wizard of Oz study to respond to the user's requests more quickly, helping to refine application behavior based on user feedback.

Background

The design of the Graphical User Interface that we created to be used as the test application within Wizard of Oz training was based on the Java language design package. The fact that the Java is a simple, robust, object-oriented, and portable programming language provided the GUI with optimum functionality. The GUI is built on the basis of objects; actions must be performed on these objects, in our case the lights, and we must have these objects work in unison. Java allows me to simulate this relationship between the lights in the lab. In Addition, realizing that my GUI would be used to collect data in a Wizard of Oz study, and that there was a definite need for my program to be moved at ease or relocated, was a very important factor in using Java as the designated programming language. Java is plat-form independent, which simply means that it has a unique ability to move easily from one computer to the other.

The GUI that I implemented utilizes the features and functionality of the Java Communications 3.0 Application Programming Interface to communicate with the lights in the lab. The API allowed me to create several other functions to set the light intensity, turn on/off the light intensity (setLightIntesity), and show the current level of the lights (showlights). "The Java Communications 3.0 API is a Java extension that facilitates developing platform-independent communications applications for technologies such as Smart Cards, embedded systems, and point-of-sale devices, modems, display terminals, and robotic equipment." [3] The Java Communications API (also known as javax.comm) provides applications access to RS-232 hardware (serial ports) and limited access to IEEE-1284 (parallel ports), SPP mode.

Methodology

From the data that was outputted by the lights on the Communication Ports I was able to add the relevant light information into vectors that were segmented according to the Communication Port, Light Group, and Light Location. The light within the lab are currently being controlled with the command line serial port interface. Communicating with the lights within our lab requires understanding how the lights were represented in terms of its real world location within the command line for the serial port interface. The HyperTerminal was used for the interface to communicate with the lights for their designated Communication Port Identification Numbers. The challenge was to identify individual lights and the characteristics of these lights that included the light intensity, location, value and current state within this vector space. The readings that I received after sending the corresponding light commands, values, and intensity level were then saved to corresponding light groups, and those groups were saved as a separate instance so that we could retrieve the unique values for that group in question. The lights in the lab are implemented with two home lighting appliance systems. X-10, a communication language technology allows the general on and off functionality of the lights within the lab. The X-10 standard for sending control signals over power lines is a relatively low-cost lighting system implementation for automated light controls. The lights themselves however are using the other system that is based on the hexadecimal unit base numbering system to represent the light location and current state of the lights throughout the lab. The light values range from lowest value (0), to highest (ff). Once the lights in the lab within the Light Control System are segmented by their perspective Communication Ports, I can see the visible state of each light within that group, and manipulate their light intensity values at any given moment.

Figure 3: X-10 Control Remote used for lights

Challenges

Developing algorithms to detect which light is on at any given point in time, send the appropriate commands for the light that is in question and having the user tell the system what task to complete is a very demanding task. During the initial design of the methods to calculate the light intensity levels, location within the light system in the lab, and the perspective location of the lights in my GUI in respect to its real-world location, we did not include check cases to monitor the type of data that were being sent into the methods. The reasoning behind this was based on the fact that the GUI in the Wizard of Oz test would be sending integer values instead of hexadecimal base values for the representation of the light values. During many stages of design methodology and code refinement, we included check cases for any and all type conversions, opening of Communication ports that the lights were located on, and error handlers for null objects and vectors containing null light values. We looked at each cluster of lights, and found similarities in the first and second arguments for each cluster's hexadecimal value. Grouped clusters according to their Identification (Id) number into vectors. (See Figure 4) The first column represents the top vector element of the first cluster, and the second column represents the bottom vector of the first cluster. Vector spaces can be scaled and added, so we created an algorithm to determine within the vector space which index would representing the current light being searched for or set to index in the vector in aims of setting and getting the corresponding light value for that individual cluster. After segmenting is completed, that is after the lights are grouped into their corresponding clusters, we can then set each light's corresponding value, location to look for, and model its real world location into the GUI to be used in the Wizard of Oz study.

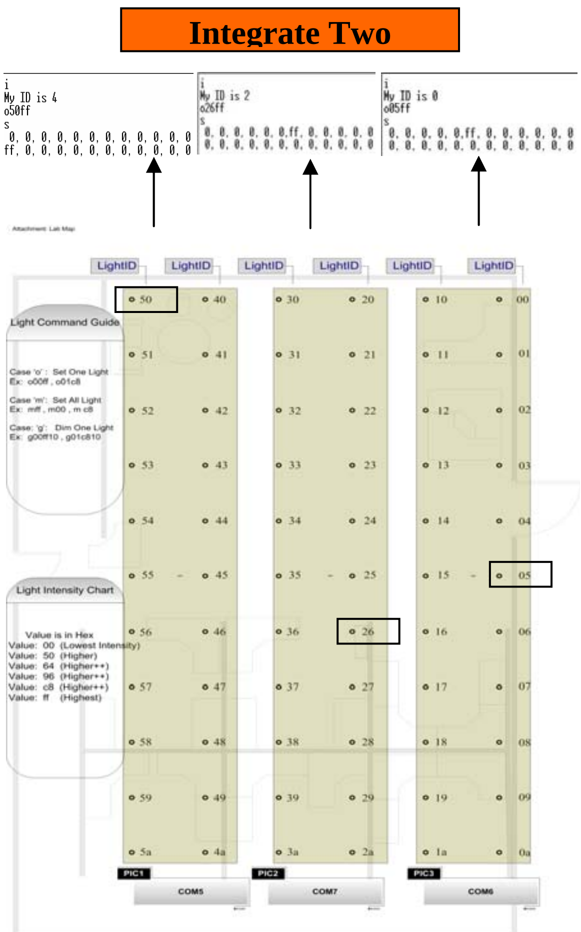

Figure 4: Integrating Two Systems

In order for our GUI and Light System in the lab to become integrated, I created several Java Classes that would maintain data abstraction. The classes served to encapsulate as closely as possible all related methods, functions, and fields all in support of having both systems to communicate with each other efficiently. Including encapsulation methods within my code, allowed me to easily maintain and modify the code. The java methodologies enabled me to group the pieces of data that describe entities that were in common, allowing me to manipulate the data as a whole. The java packages that I created are based on two main methodologies, the first was designed to communicate with the Communications Port in Windows, and the second was designed to maintain the general look and feel of the GUI that served as the desktop representation of the lights within the lab. These two design methodologies were then broken down into more subclasses and sub-methods as we refined the project. (See Figure Class)

Class	***Function***	***Purpose***	***Result***
CommLight .java	CommLight()	Fills in light location and cluster group according to light being looked at	Places Light being looked at in respective cluster.
CommLight .java	getComdID()	Returns the ComID for the current light being looked at	ComID
CommLight.java	getLightLoc()	Returns the current location of light beign looked at	LightLocation

Figure 5: Figure of Java Classes

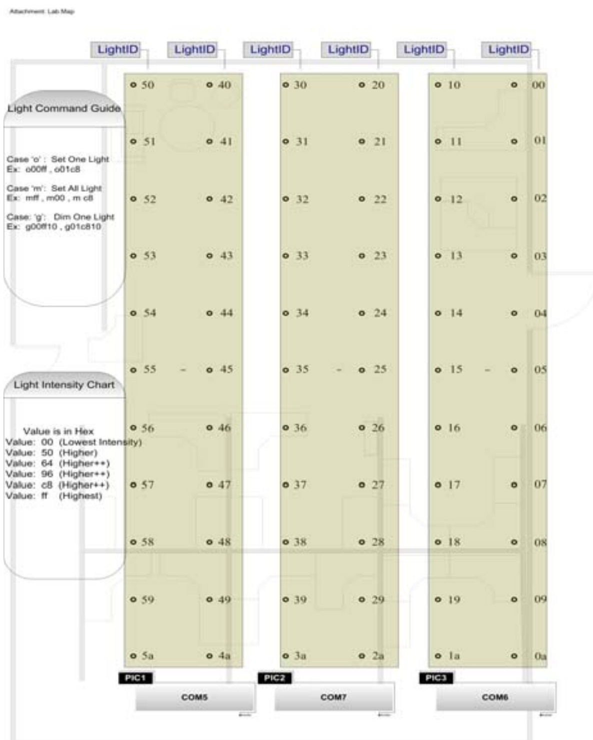

Figure 6: Mapping GUI with HyperTerminal

Evaluation

The design process of the future speech interface will require us to study and collect relevant data that will measure the level of usability for the GUI. We performed a Usability Evaluation Test on the software tool that I created for use in the wizard-of-oz study. The Usability Evaluation Test provided us with an accurate indication of how well users were able to understand and use the GUI. Moreover, the test gave us a gage on how effective each system of communicating with the lights was, and how quickly the participant would respond to my requests for the manipulation of the lights. The Usability Evaluation Test was performed with 4 participants (wizards), where each participant was asked to perform 8 tasks for manipulating the lights in the room. The tasks collectively, were performed on both the command line interface for the serial port (HyperTerminal) and the GUI. The task required the participants to perform 8 basic light manipulation functions such as, "Turn on Lights over Public Machine" , Turn on One Light Over David's Cubical" , "Show Current Light Intensity for lights over Ana's Cubical". Each task was timed from the time they started to perform the action until the time they completed the task on each system for communication with the lights. Each participant in the test was led to their own cubical where each individual participant evaluated the interface alone. The test-application was set up on one computer, thereby maintaining the privacy of the participant. The reasoning for having each participant inspect the GUI alone was to ensure that there were no biased or dependent evaluations from the participant communicating with each other.

The User Evaluation Test illustrated unequivocally, that the GUI was much easier to use than the command line interface. The test also provided me with ideas for future improvements to the GUI that would aid in its effectiveness.

Figure 5: Descriptor Class to Calculate Light Group and Location

Results

In General, the participants in the usability study reported that the GUI was easier to use for all features, that it was more visual, and it was a better representation for the real-world location fixtures of the lights throughout the lab. Some participants, however found that the HyperTerminal in some cases were a little bit easer for performing dimming operations on the lights. Participants collectively shared similarities in their belief that the GUI was a better real-world representation for the light locations in the lab and for manipulating the lights by turning them on and off. The fact that the lights were grouped in a visible state allowed the participants to easily manipulate a group of lights at once.

The GUI was by far, more usable for the participants in comparison to the HyperTerminal system. (See Figure 7) Participants provided many insights into the future improvements for the GUI to control the Light System in our lab; they felt that there should be some kind of visual, onscreen element that can show the current state of the light. Also, they agreed that the efficiency and functionality of the GUI would be improved if a multiple selection highlighting system was included to allow the manipulation of multiple lights. The goal that I had in increasing the speed in the response time from a wizard in a Wizard of Oz Study was achieved with the creation of the GUI, the GUI allows the wizard to analyze user utterances more quickly than the command line interface. (See Figure 8) The participants evaluation of the GUI, will allow future developers to build and include improvements and refinements upon the GUI. From the Usability Test, we can infer that future improvements made to the GUI will allow the wizard, in a Wizard of Oz study, to more quickly respond to user's requests, allowing him to closely simulate an implemented speech interface for the lighting in the lab.

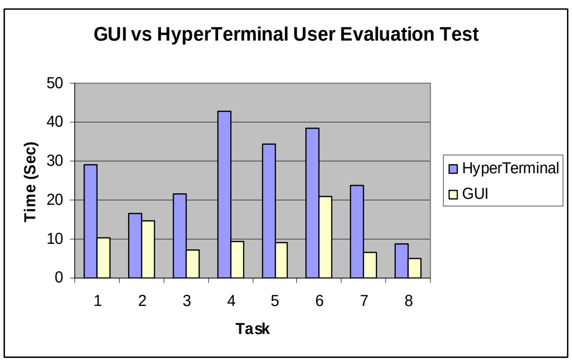

Figure 7: Usability Evaluation Test

HyperTerminal User Evaluation Test	Time (Seconds)		GUI User Evaluation Test		
Task	Avg.	Std. Div	Task	Avg.	Std. Div
1	29.2	13.34241357	1	10.225	2.625991876
2	16.575	9.478528367	2	14.7	5.337914699
3	21.65	8.174146235	3	7.075	3.140461749
4	42.925	21.86220102	4	9.35	3.767846423
5	34.4	15.5385542	5	9.2	1.779513042
6	38.55	10.21779493	6	21	18.08258831
7	23.85	10.12011199	7	6.5	0.867947771
8	8.7	3.024345659	8	5.125	1.719253714

Figure 8: Usability Evaluation Test (Standard Deviation)

Conclusion

The GUI will enable the wizard in the study to quickly react to user's requests to manipulate the lights in the lab, thereby helping us to simulate an implemented speech interface. The Wizard of Oz study will provide an accurate research tool for designing realistic dialogue between humans and speech interface systems. The next step in the development and advancement for the GUI will be to possibly add multiple selection group functionality where the wizard where the wizard can select multiple groups with one click and manipulate them to his wishes.

Future Work

Because of GUI used in this study, we will be able to perform research necessary for modeling and designing the future speech interface system. The next step in the development and advancement for the GUI will be to possibly add multiple selection group functionality where the wizard can select multiple groups with one click and manipulate them to his wishes. The lights currently are manipulated by using a dimming system, where as the actual turn off/on are performed by the X-10 light appliance system, a possible improvement would be to integrate the GUI with the system to turn on/off the lights completely.

Acknowledgements

Working and conducting research in BID has been a wonderful experience. I would like to express my great appreciation to my faculty mentor Dr. John Canny for allowing me use my intelligence and skills to explore and examine the research in HCI, graduate mentor Ana Ramirez Chang for her guidance, the many students in BID, and SUPERB-IT sponsored by the National Science Foundation.

References

1. Fraser, N.M. and G.N. Gilbert, "Simulating Speech Systems," Computer Speech and Language, Vol. 5, Academic Press Limited, 1991. <http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide/UserInterface.html>.

2. A.Cavalluzzi, F.de Rosis, I.Mazzotta and N.Novielli. Modeling the user attitude towards an ECA. UM'05 Workshop on "Adapting the Interaction Style to Affective Factors". Edinburgh, July 2005. http://www.di.uniba.it/intint/Humaine/H-Woz.html#publications
3. Sun Microsystems, Inc. "Java Communications API." Sun Microsystems, Inc. JCP. July 28, 2006. <http://java.sun.com/products/javacomm/>.
4. Nils Dahlback, Arne Jonsson, Lars Ahrenberg. "Wizard of Oz Studies - Why and How." Department of Computer and Information Science Natural Language Processing Laboratory (1992): 8.
5. Jose F. Quesada, Federico Garcia, Esther Sena, Jose A. Bernal, Gabriel Amores. "Dialogue Management in a Home Machine Environment: Linguistic Components over an Agent Architecture." Universidad de Sevilla Grupo de Investigacion Julietta (2001): 9