Using Dialog and Context in a Speech-Based Interface for an Information Visualization Environment

Kenneth Cox, Rebecca E. Grinter
Stacie L. Hibino, Lalita Jategaonkar Jagadeesan
Bell Labs, Lucent Technologies
263 Shuman Boulevard
Naperville, IL 60566 USA
{kcc, beki, hibino, lalita}@research.bell-labs.com

David Mantilla
Department of Computer Science
Harvard University
Cambridge, MA 02138 USA
(work conducted at Lucent Technologies)
mantilla@fas.harvard.edu

ABSTRACT

We describe a speech-based interface to an information visualization (infoVis) system. Users ask natural-language questions about a given data domain. Our interface then maps the questions into infoVis operations, which result in the display of data visualizations that address the questions. Users can interact with these views via speech or direct manipulation. If users give incomplete information, our interface guides them in clarifying their questions. The intelligence behind our interface is encapsulated in a service logic that embodies domain knowledge about both the data being explored and the infoVis system. This allows users to focus on answering questions, rather than on the mechanics of accessing data and creating views.

Keywords

information visualization, speech and natural language interfaces, multi-modal interfaces

1. INTRODUCTION

User interaction with infoVis frameworks is typically limited to the mouse and keyboard. Advances in automatic speech recognition make speech-based modes of interaction possible. While we could build a speech interface that mimics the mouse interface, user studies suggest that speech is not effective when used in this way [3]. Instead, following previous research recommendations [4], we take advantage of the inherent features of speech to provide a dialogue interface rich with domain knowledge.

In this paper, we present a novel approach to adding a speech interface to an infoVis framework to aid users in conducting data analysis. One of our goals is to integrate multiple interaction modes, including natural language, both to increase accessibility and to enrich users' experiences. To achieve this, we allow users to easily move back and forth among various interaction modes, according to their preference.

2. SYSTEM DESCRIPTION

2.1 Approach

Our approach focuses on the use of a service logic for mapping domain-specific inquiries about data into infoVis operations. These operations create database queries and present views for interactively investigating a given inquiry. This provides higher-level support for infoVis, by using domain knowledge about specific data sets. In particular, it aids users in exploring data by using details about the syntax and semantics of the data, as well as knowledge about infoVis capabilities. This approach allows users who have little knowledge of the structure of the data set and/or infoVis tools to work with the data more easily and simply.

2.2 Design

Our architecture, shown in Figure 1, combines three powerful components: the *IBM ViaVoice* speech recognition system, the *Sisl* (several interfaces, single logic) architecture for creating services with multiple user interfaces [1], and the *InfoStill* infoVis framework for helping users distill information from data [2]. The Sisl service logic uses a natural-language user interface implemented with IBM ViaVoice and interpreted with input from the Sisl speech grammar to send commands to InfoStill. Commands sent to the Presentation Manager (PM) control and select visualization views, while commands sent to the Task Manager (TM) perform database queries and create views to present to the user.

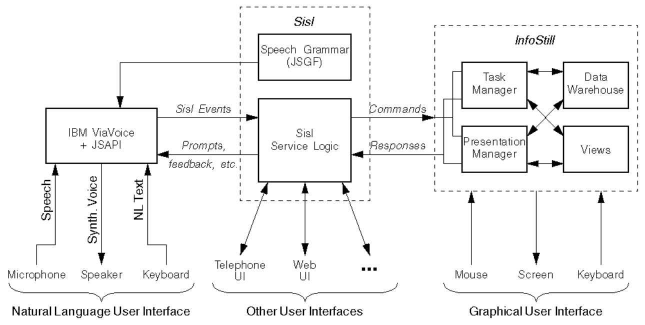


Figure 1. Overall System Architecture.

The intelligence behind our speech interface to InfoStill is encapsulated in Sisl. The Sisl speech grammar is used to specify the user utterances that are recognized by the system. The Sisl service logic, based on reactive constraint graphs, is used to manage a dialogue with the user to refine a vaguely worded natural language inquiry into a well-defined database query and then translate this query into one or more InfoStill commands.

Our system is built in Java 1.1 and uses IBM ViaVoice and its associated Java Speech API library on a PC.

3. EXAMPLE APPLICATION

In our first application, we built a speech-based interface to InfoStill for analyzing organizational data. Specifically, we wrote a Sisl grammar and logic to allow our system to visualize Lucent Technologies' personnel database. This database contains information such as the department, manager, physical location, job title, and phone number of every employee.

We interviewed organizational experts (who are not infoVis experts) and identified threads of inquiries that we used to demonstrate the power of domain-specific natural language interfaces and multi-modal interaction. For example, suppose the user poses the following question:

User: How has [department x] grown over time?

The Sisl logic recognizes that the query is only partially specified and prompts for missing temporal information:

Service: As of when?

User: As of now.

The Sisl service then determines that the query has been completely specified and sends appropriate TM commands to InfoStill. Two of the generated views appear in Figure 2.

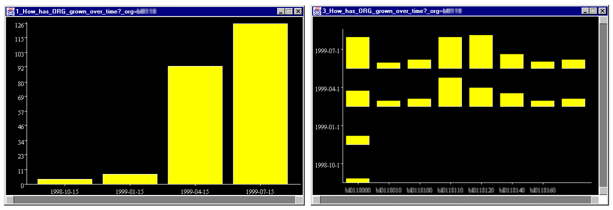
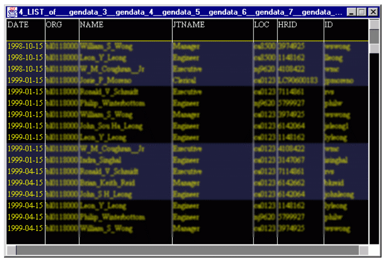


Figure 2. The left bar chart shows the total number of people in 'department x' for four time periods. The right grid chart divides 'department x' by groups, displaying the group names on the x-axis and time periods on the y-axis.

The user might now be interested in seeing the actual names of the employees and so might say:

User: Show me details for the bar chart.

An interactive values list (spreadsheet-like) view of data items is then displayed as shown in Figure 3. The values list is linked to the bar chart and grid chart in Figure 2 so that selections made in one view automatically propagate to all other views.



4_LIST_of___gendata_3__gendata_4__gendata_5__gendata_6__gendata_7__gendata_...

DATE	ORG	NAME	JTNAME	LOC	HRID	ID
1998-10-15	hl0118000	William_S_Wong	Manager	ca8500	3974925	wswong
1998-10-15	hl0118000	Leon_Y_Leong	Engineer	ca8500	1148162	lleong
1998-10-15	hl0118000	W_M_Coughran__Jr	Executive	nj9620	4108422	wmc
1999-01-15	hl0118000	Josie_P_Moreno	Clerical	ca0123	LC90600183	jpmoreno
1999-01-15	hl0118000	Ronald_V_Schmidt	Executive	ca0123	7114861	rvs
1999-01-15	hl0118000	Philip_Winterbottom	Engineer	nj9620	5799927	philw
1999-01-15	hl0118000	William_S_Wong	Manager	ca0123	3974925	wswong
1999-01-15	hl0118000	John_Sou Ha_Leong	Engineer	ca0123	6142064	jsleong1
1999-01-15	hl0118000	Leon_Y_Leong	Engineer	ca0123	1148162	lyleong
1999-01-15	hl0118000	W_M_Coughran__Jr	Executive	ca0123	4108422	wmc
1999-01-15	hl0118000	Indra_Singhal	Engineer	ca0123	3147067	isinghal
1999-04-15	hl0118000	Ronald_V_Schmidt	Executive	ca0123	7114861	rvs
1999-04-15	hl0118000	Brian_Keith_Reid	Manager	ca0123	6142662	bkreid
1999-04-15	hl0118000	John_S H_Leong	Engineer	ca0123	6142064	johnleong
1999-04-15	hl0118000	Leon_Y_Leong	Engineer	ca0123	1148162	lyleong
1999-04-15	hl0118000	Philip_Winterbottom	Engineer	nj9620	5799927	philw
1999-04-15	hl0118000	William_S_Wong	Manager	ca0123	3974925	wswong

Figure 3. Corresponding values list view.

For example, if the user says:

User: Select the largest bar in the bar chart.

The rightmost bar of the bar chart is selected and the other two linked views are also updated. The user may continue to interact with the views using voice or direct manipulation; for example, the user may click on the mouse to hide unselected data. Alternatively, the user may begin a new line of inquiry.

4. DISCUSSION AND FUTURE WORK

Adding speech-based interfaces to InfoStill has several benefits. First, the Sisl architecture provides contextual dialog that is capable of handling both partial and multiple commands. This allows users to give a series of questions or commands, without having to wait for each to be processed. Moreover, users can speak any combination of full and partial phrases while the system maintains the context. For example, a user can say "Select bar 1," see the results, then say "and 4" to add bar 4 to the selection.

A second benefit of our approach is that we do not replace the GUI with a speech interface, but rather support multi-modal interaction where users can seamlessly move between the modes. Finally, combining Sisl and InfoStill provides a modular and extensible design. We can update Sisl speech phrases without changing InfoStill, and similarly can extend InfoStill with new commands or views without restructuring the existing Sisl grammars and logic.

Overall, our approach empowers users to explore data in a way that has not previously been possible, guides them in their explorations when they need help, and allows them to choose the most natural mode of interaction for exploring data within an infoVis framework. We plan to apply our approach to other data domains and conduct a user study to compare the usability and utility of the speech-enabled version of InfoStill to the standalone one. We also plan to compare our approach to that of "wizards" and similar systems that assist users with complex tasks.

REFERENCES

[1] Ball, T., Colby, C., Danielsen, P., Jagadeesan, L., Jagadeesan, R., Laufer, K., Mataga, P., Rehor, K. Sisl: Several Interfaces, Single Logic. To appear in the *International Journal of Speech Technology*, Kluwer Academic Publishers.

[2] Cox, K., Hibino, S., Hong, L., Mockus, A. and G. Wills. (1999). "InfoStill: A Task-Oriented Framework for Analyzing Data Through Information Visualization." In *IEEE Symposium on Information Visualization Late-Breaking Results*.

[3] Damper, R. and S. Wood. (1995). "Speech versus keying in command and control applications," *International Journal of Human-Computer Studies, 42*, 289-305.

[4] Hugunin, J. and V. Zue. (1997). "On the Design of Effective Speech-Based Interfaces for Desktop Applications." In *EuroSpeech'97 Conference Proceedings.*