GENERIC VOICE INTERFACE FOR COCKPIT APPLICATION

David T. Williamson
Gregory L. Feitshans

Air Force Wright Aeronautical Laboratories (AFWAL/FIGR)
Wright-Patterson Air Force Base, OH

ABSTRACT

Voice technology for cockpit application is advancing rapidly. Steady improvements in algorithms and hardware packaging are yielding systems with unprecedented capabilities. Being able to take full advantage of these new capabilities, however, still remains a problem. Application development tools typically lag behind hardware improvements making it difficult for novice users to successfully apply this technology. This paper discusses a Generic Voice Interface that would allow both novice and expert users of voice input and output devices to quickly develop and integrate their applications while still achieving optimum performance.

INTRODUCTION

The capabilities of currently available voice recognition devices have increased dramatically over their predecessors. Larger vocabularies and connected speech operation are just a few of these improved capabilities. While the basic hardware and recognition algorithms have improved, there still remains the problem of how to take full advantage of these improvements. Each system has its own unique way of defining vocabularies, training the vocabularies, building a syntax structure, and finally interfacing the device to an application. This series of tasks is not usually easy, even for the more experienced user. The application development software provided with these systems has lagged far behind the technology developments. This lack of good application development tools often results in a degraded voice interface which prevents a fair assessment of the technology.

To overcome this lack of development tools, the Flight Dynamics Laboratory has been actively involved in the creation of several software programs that simplify the use of off-the-shelf voice recognition systems. These include tools for creating vocabulary and syntax structures, automated training, and data analysis. Although these tools have been successfully applied in various simulation and flight test programs, they are limited because they were created for specific voice systems. As new systems become available, it would be highly desirable to apply generic speech tools that would be independent of device specific software requirements.

This paper describes a voice technology interface that would allow both novice and expert users of voice input and output devices to quickly interface them to their applications while maintaining optimum performance. The key feature of this Generic Voice Interface (GVI) will be to provide a device independent interface to existing voice systems.

OVERALL CONCEPT

The GVI is broken into modules according to the tasks to be performed. This allows portions of the GVI to be device independent and thus portable to other voice I/O devices. Figure 1 shows the overall concept for the GVI system. The GVI uses a task-based approach to describe the application environment. The typical development sequence would be as follows. First, the application developer identifies a series of tasks in the Task Description Module (TDM). Each of these tasks are entered into the GVI with other information specific to the task. A selection of words or phrases to activate a given task are also entered here. Next, the user has the ability to customize the voice interface via the Application Tailoring Module (ATM). After the user has defined a desired task domain, the Vocabulary Training Module (VTM) provides the necessary routines to automatically train and optimize the voice system to the user's voice characteristics. The specific requirements of the voice system are translated in the Voice I/O Interface Module (VIM). Finally, the Application Interface Module (AIM) provides the necessary information management between the voice system and the outside world to allow the pilot to successfully use voice as a cockpit control and display technology. A more detailed description of each of these modules is discussed below.

TASK DESCRIPTION MODULE

The Task Description Module (TDM) allows the application developer to describe the actual tasks that can be completed by voice. The tasks can be grouped by category in order to provide for logical task selection by the user. Each task will contain a description of how the task works, what it accomplishes, and various choices of words that may be used for accomplishing the task.

U.S. Government work not protected by U.S. copyright.

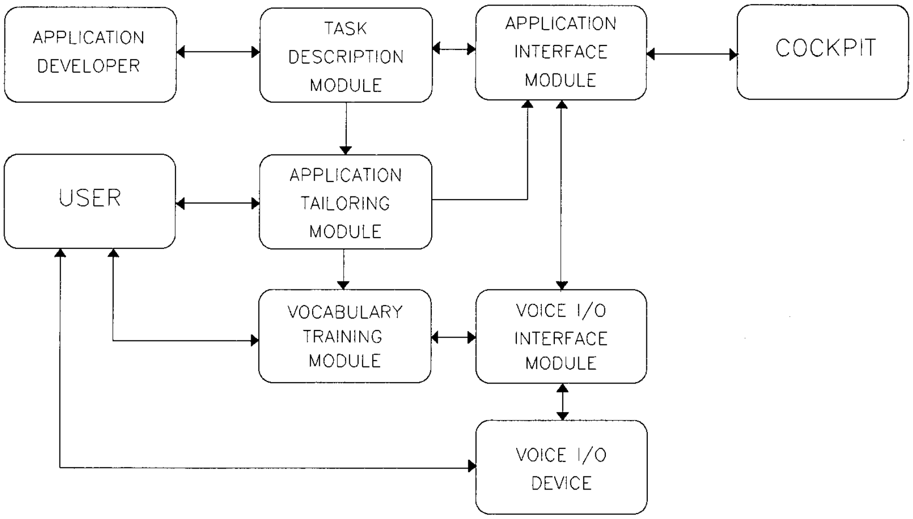
APPLICATION DEVELOPER
TASK DESCRIPTION MODULE
APPLICATION INTERFACE MODULE
COCKPIT
USER
APPLICATION TAILORING MODULE
VOCABULARY TRAINING MODULE
VOICE I/O INTERFACE MODULE
VOICE I/O DEVICE

Figure 1. Generic Voice Interface Concept.

APPLICATION TAILORING MODULE

The Application Tailoring Module (ATM) will present the tasks to the user in a logically grouped manner. The user will be able to move through the various tasks available, selecting only the desired ones. Tasks can be chosen by selecting the category, which takes all tasks in that category, or by selecting individual tasks within a category.

As a part of the ATM, certain words or phrases will be designated as macros. When spoken, these words will activate a series of tasks that the user selects for that word. An example for a fighter cockpit application might be "**SET UP FOR INGRESS**", which could turn on targeting radar, arm air-to-ground weapons, and turn on appropriate countermeasures.

The ATM will also contain code to enable the selection of various types of output for the different tasks. Each task will list the ways the output of the task can presented. The user chooses the method of output most desirable. The tailoring could even be such that different voices, such as robotic, female, etc., could be chosen for different tasks.

VOCABULARY TRAINING MODULE

The Vocabulary Training Module (VTM) will train the user on the selected vocabulary. The training will be as automated as possible and still provide optimum recognition performance. The VTM will accomplish two functions. While the required voice patterns are being generated, the user will receive training on the interaction of the commands to the desired cockpit control functions. This embedded training will greatly shorten the overall training process. If the training takes a lengthy period of time, breaks will be incorporated at appropriate points in the process. In addition to the training, an easy method will be developed for verifying the vocabulary and updating or retraining portions that seem to be causing problems. From this verification process, device dependent threshold settings will be automatically adjusted to provide the best overall performance.

APPLICATION INTERFACE MODULE

The Application Interface Module (AIM) monitors any parameters necessary to provide a contextual framework to assist in interpreting the voice commands. Once it has parsed the command, the result is sent to the target application. The application should then have the logic to determine what action should now take place. By monitoring various parameters the AIM will be able to determine the state of the application and be able to signal this information to the voice system, via the Voice I/O Interface Module (VIM), to change template sets or do whatever is necessary to reduce its search space.

The AIM will continually make calls to the VIM turning tasks on and off as it monitors the state of the application. When a task is shut off, the corresponding words associated with that task are also shut off in the recognition device. Each phrase or word will have a task flag associated with it. As long as all tasks for that word are not shut off the word or phrase will still be active. This approach will allow for adding words or phrases without any code changes. There will be two types of syntax management from this. First, active tasks will specify which words are currently active and second, as words are spoken within a task, movement through the syntax structure will cause various words to be activated or deactivated. The knowledge the AIM would require to properly manage the transitions through the syntax could be extensive, especially for cockpit applications.

VOICE I/O INTERFACE MODULE

The Voice Input/Output Interface Module (VIM) handles the device specific routines that are used to control the particular recognizer or voice output device being used. A module of this type would have to be created for each device being connected to the GVI. The GVI could then be used with various I/O devices by simply linking with the VIM of the system desired. The goal is to make the GVI look the same to the user regardless of the type of I/O system being used. This approach has already been implemented in the computer graphics area using the Graphics Kernel System (GKS). GKS drivers exist for graphics systems that allow programmers to create device independent graphics programs. For example, a GKS call to draw a circle of a certain radius can be made without the programmer having to be concerned about the specific graphics device. Taking this idea into the voice area, a voice device driver should be able to accept high-level calls for Train, Recognize, Save Templates, etc., and translate them into the required calls for a given voice device.

SUMMARY

The potential application for the Generic Voice Interface is limitless. The system will be designed so that any application, not just cockpit applications, can be used with the GVI. Once it has been successfully integrated into a few key applications, the same techniques can be transitioned to other areas. The system will initially be targeted for the Rapidly Reconfigurable Crewstation (RRC) program, which will provide a rapid prototyping environment for advanced crewstation design. Another targeted application will be the Super Cockpit program. Both programs can benefit significantly from the capabilities of the GVI system.

A great deal of enthusiasm has been expressed by the application development community about the GVI approach. Many users are frustrated with the lack of application development tools and with the length of time required to integrate voice systems. It is hoped that this system will solve some of these problems to allow a more painless evaluation of voice technology for their applications.