

VUI as an Alternative User Interface for Field-Based Military Applications

Brian Stygar, Principal
Kore Federal
bstygar@korefederal.com

Overview

Much of the software used in business today takes the form of complex graphical user interfaces (GUIs). GUIs allow users to perform many tasks simultaneously while maintaining the context of the rest of their work. These interfaces, though, are often mouse and keyboard intensive which can be problematic in some work environments that require heavy use of hand and eye coordination, such as military battlefield settings.

This is especially the case where mobile applications are used to support numerous tasks involving real-time utilization of databases or documents. Mobile military battlefield applications such as tactical battle command systems, field artillery tactical data systems, command and control systems, intelligence and maneuver control systems often require a hands free user interface, which enables the use of the hands for other critical tasks. An excellent substitute for the mobile application GUI in these cases is the Voice-enabled User Interface, or VUI.

The VUI can offer many benefits over the traditional GUI. First and foremost, voice as a user interface makes it possible to interact with a mobile application without having to hold any peripheral "devices". So, speech input does not interfere with manual tasks, such as executing fire support or air defense operations.

In a similar fashion, advances in text-to-speech (TTS) technology enable text information to be transmitted easily to the user. With TTS, mobile application users can receive query responses, hands-free and eyes-free.

Finally, speech is the natural method most often used when communicating with other people. Fundamentally, this should hold true for human-computer interaction. It should be easier for a user to learn the operation of voice-activated commands and responses.

Business Need

Mobile military applications could really benefit from a Voice-enabled User Interface. These applications are primarily used by military forces, commanders and their staff in the field executing mission-related combat support activities. Their eyes and hands can be preoccupied while completing manually intensive tactical operations exercises. The modern battlefield must also endure extreme environmental conditions, where it is difficult to protect even the most rugged mobile computing devices.

A VUI solution could improve the efficiency of the battlefield or training application standard processes, and improve the quality of data entry. With the use of a wireless microphone and headphone set, the mobile devices themselves could be contained and protected from the elements.

Solution Landscape

A Voice-enabled mobile military application would require the integration of several emerging technologies. Both hardware and software components must be employed to create a hands free, voice controlled interface to make the application multi-modal. The following technologies are necessary:

- 1) Automatic Speech Recognition (ASR)
- 2) Text to Speech (TTS)
- 3) VoiceXML
- 4) Speech Software Development Kits (or Speech APIs)
- 5) Bluetooth Headsets

Basically, the Automatic Speech Recognition component allows the computer to identify the words that a person speaks into a microphone. It provides the dictionaries and processing to convert continuous speech into text. ASR also incorporates the training programs to improve the recognition of a users speech patterns.

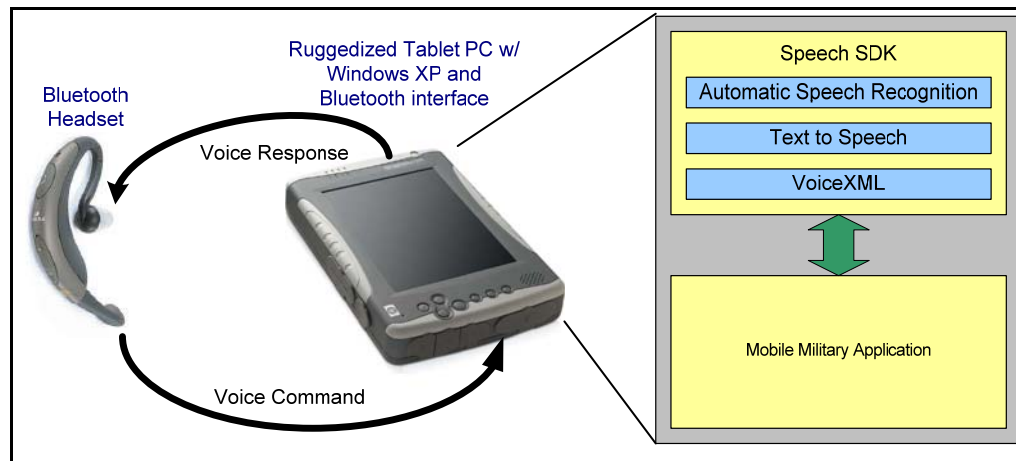
Computer responses are communicated back to the user via the Text to Speech (TTS) component. The TTS synthesizer uses Digital Signal Processing (DSP) and Natural Language Processing (NLP) technologies to convert text to natural, spoken word.

Voice Extensible Markup Language (VoiceXML) is a markup language for creating voice user interfaces that use automatic speech recognition (ASR) and text-to-speech synthesis. VoiceXML provides the standard message interface between the Speech SDK and both the ASR and TTS components.

All of the above technologies need to be wrapped by a windows-based Speech Software Development Kit (SDK) or API so that the mobile military application can interface with the capabilities of the voice/speech processors. The Speech SDK must support an array of Microsoft .Net and J2EE languages, and standard message formats.

Bluetooth headset provides a wireless connectivity between the earbud/microphone, worn by the user, and the Tablet PC device. Both the headset and PC device must support the Bluetooth standard. Communication between the end user and PC hardware is transmitted wirelessly over a short range to create a true hands free environment.

All of the above technologies are described in detail in the next section. The following figure describes how these technologies integrate together to provide the VUI solution architecture.



Design Considerations

The solution cannot be complete without consideration for proper VUI software design. Voice commands must be created that are easy for a user to remember. The interface should be easy to use and require minimal user attention. The VUI also needs to consider application navigation, command initiation, data entry and basic information retrieval/reply. Finding and working with distinct transactions, following links, entering data, and activating buttons is much more difficult in a Voice controlled system than a traditional GUI.

The VUI design will need to consider the usability trade-offs between user efficiency and ambiguity handling. Basic interaction should be able to be performed using high-level commands, as opposed to a more natural voice interaction (e.g. search for a record by key name).

Technology Components

The following technology components would be used in the mobile military application Voice-enabled User Interface solution:

- 1) Automatic Speech Recognition (ASR)
- 2) Text to Speech (TTS)
- 3) VoiceXML
- 4) Speech Software Development Kits (or Speech APIs)
- 5) Bluetooth Headsets

These individual technologies are described in detail in this section.

Automatic Speech Recognition (ASR) Software

Automatic Speech Recognition (ASR) Software, also known as Voice Recognition Software (VRS), speech recognition, or natural language recognition software, essentially converts voice to text on a computer. When a user speaks through a microphone connected to a PC with ASR, the software "translates" the sounds into written words in your application of choice. While the first-generation ASR software packages implemented discrete speech technology, where you had to pause between words in order for the computer to understand them, ASR software now uses continuous speech technology, which allows you to speak more naturally.

The two key characteristics of speech recognition are accuracy and fluency. Accuracy is the aspect of the technology that allows it to correctly identify each word you speak as a word it knows. Fluency is the measure or range of grammar and vocabulary that the technology knows.

ASR software packages should have the following features and capabilities:

- Large vocabulary of at least 150,000 words and the ability to add at least 50% more words.
- Integration with many common software integration architectures and standards.
- The ability to use natural language phrases.
- Flexible and powerful training software that allows the software to recognize the way you speak.

There are many commercially available ASR software solutions in the marketplace. There are really only two main competitors who offer a product that meets the feature requirement listed above: IBM and ScanSoft. The following table lists a few of the popular packages available today.

Vendor	Product	Strength
IBM	ViaVoice	Best of Class
ScanSoft	Dragon NaturallySpeaking	Best of Class
Microsoft	Windows XP Voice Recognition	Integration w/ Windows
Commodio	QPointer	Shareware
Ultimate Interactive Desktops Inc	Voice Studio	Shareware
Realize	Voice Lite	Shareware

Text-to-Speech

Text-to-speech software is used to turn a text string into spoken language. This results in the ability of the computer to talk to the user, saving the user from having to read the text on the screen.

Text-to-speech software consists of a speech synthesizer and a set of rules for translating text to input strings to the speech synthesizer. The translation from text-to-pronunciation is

also central to a full text-to-speech system. The best TTSs offer natural-sounding, highly intelligible text-to-speech synthesis.

Key features of robust TTS software include:

- Fully compliant with the Industrial Standard Microsoft Speech API (SAPI)
- Multilingual support – read source text of multiple languages
- Accent and dialects creation
- Control flexibility - ability to set volume level, rate and pitch of synthesized voice
- Supports a full set of VoiceXML Speech Tags
- Automatically detects and processes numbers, time, dates, URLs, e-mails, phone numbers, contractions, abbreviations

Similar to ASR, there are many commercially available TTS software solutions. The following table lists a few of the popular packages available today.

Vendor	Product	Strength
IBM	WebSphere Voice Server	Best of Class
Marsiansoft	TextSpeech Pro	Best of Class
ProNexus	VBVoice	Integration w/ Windows
NaturalReader	NaturalReader	Shareware

VoiceXML

VoiceXML is a standards-based markup language for creating voice-user interfaces. It uses speech recognition for input, and text-to-speech synthesis (TTS) for output. It leverages the web paradigm for application development and deployment. By having a common language, application developers, platform vendors, and tool providers all can benefit from code portability and reuse.

VoiceXML replaces Interactive Voice Response (IVR) technology with a W3C standard markup that offers reusable and off-the-shelf applications, making it three times faster and less expensive than traditional IVR. VoiceXML basically enables integration of voice services with data services using the familiar client-server paradigm. Its greatest strength is that it promotes service portability across implementation platforms. VoiceXML is a common language for content providers, tool providers, and platform providers.

Speech SDK

Microsoft offers a comprehensive Speech SDK for building speech-based applications and interfaces on the Windows platform. The SDK contains a collection of speech-oriented development tools for compiling source code and executing commands including the Win32-compatible speech application programming interface (SAPI), a continuous speech recognition engine and a text-to-speech engine.

Bluetooth Headsets

Bluetooth headsets offer a convenient and user-friendly wireless connection between a headset and a Bluetooth-enabled device such as wireless phones, PCs, cameras, GPS devices, and handhelds. The headsets themselves have a mic boom and small ear speaker. They can sit on the ear or wrap behind it.

Bluetooth provides a way for different devices to communicate with each other by sending data via a secure, short-range radio frequency. Bluetooth allows up to seven connections to be made at one time, at a speed of 1Mbps.

Feasibility

Developing a VUI based solution is quite feasible, given the availability and maturity of the solution components as described previously. The greatest challenge will be truly leveraging the capabilities of these emerging technologies through a well designed and integrated user interface. For instance, the ASR component can easily receive and translate voice commands into interface messages; however, can the UI be designed in such a way that navigation and data entry is easy and efficient for the end user.

Further, the TTS component can take a query result and synthesize it into a voice response for the user. What if the result involves multiple records with multiple fields of information? It could intimately result in an interface that is more cumbersome and difficult to use than the traditional GUI, unless considerations are made toward usability in the new voice paradigm during interface design.

Risks

The major risks of the Voice-based User Interface are based on the interface design as opposed to the integration of the solution components. It is difficult to determine whether or not a suitable voice interface can be designed in such a way that maintains the original intention and effectiveness of the GUI.

For instance, with a VUI there is an inherent challenge in specifying focus in an efficient manner. What happens in the case where there are multiple selectable entries with the same label, such as an option control? Both the application and the user need to know the current focus and when a command is issued; and spoken commands can be ambiguous. With a VUI, users do not have a way to resolve ambiguities physically (as with a mouse move and click); therefore, either the ambiguity must be prevented somehow or another method must be used to perform the resolution. In most cases, the system needs to be able to help the user resolve possible ambiguities and to help users keep track of what they are doing.

Efficiency in VUIs speech interfaces is often lost when transitioning from a GUI. It will usually take longer to select a target by voice than to point and click; or to listen to the computer dictate a paragraph of information, than to read it off of the screen. To solve these issues, VUIs increase efficiency by limiting the number of commands/responses and keeping them short.