Voice-Enabling Enterprise Applications

J. Chugh, V. Jagannathan
jandeep@careflow.com, juggy@careflow.com

Abstract

This paper reviews the influence of the advancement and standardization of Voice technologies on existing web and enterprise applications. We focus on the limitations and concerns of a new voice interface, in addition to the integration issues. We use automation of the dictation process, which is usually a part of enterprise transcription applications, as a case study to explore the power of current voice technologies. We specifically delve into VoiceXML and also into the fundamental integration solutions.

1. Introduction

Phenomenal growth and standardization in the telephony voice market and speech recognition technologies has driven the need for enhancing the existing enterprise applications running on the visual web. However, there exist traditional and popular IVR (Interactive voice response) solutions that are not portable, flexible and are hard to develop as they run on propriety platforms. And the biggest drawback is not being able to leverage existing vast web infrastructure. Speech Web, a combination of voice and data standards woven together with XML and other technologies has presented us with new competent choice. But integrating speech web with enterprise/web applications involves much more than simply adding a new voice interface to an existing internet/intranet based solution. Simply, because the users have different sets of needs and expectations for voice applications as opposed to visual ones. Users want voice transactions that are reachable at any time, easy to use, tailored specifically for their requirements, robust fault tolerant and prompt in responding. VoiceXML is one such technology available to satisfy these needs. We use automation of dictation process, as a case study to briefly explore the various capabilities (such as barge-in, recording, security etc.) of current voice technologies, specifically VoiceXML and also to delve into the fundamental integration issues.

2. VoiceXML and example Application

VoiceXML plays the role of the language of communication in voice applications, similar to the role played by HTML in web applications. VoiceXML is a member of the XML family. It is a W3C specification that bring together number of useful capabilities, such as: automatic speech recognition, touch-tone keypad recognition, text-to-speech playback, pre-recorded audio playback, telephony features, dialog features, event handling, scripting, platform features and performance features. These capabilities can be employed to develop rich user experiences that allow callers to access information and transaction services through a telephony device.

The architectural design of VoiceXML is very flexible and integrates seamlessly with the existing web technologies. As shown in figure 1, the general model of voice application architecture comprises a VoiceXML gateway or server that sits between the wired / wireless phone network and the HTTP server and is primarily responsible for actual translation process between voice and VoiceXML contents.

Figure 1. Voice Application Architecture

Figure 1 illustrates a simplified architecture of Dictation application system. It automates the dictation process workflow using VoiceXML that is integrated with existing CORBA based Careflow’s transcription server. As the part of transcription workflow, the physician has been provided with voice user interface to give and review dictations. The middleware for voice services generate dynamic VoiceXML contents by collaborating with different sources of information and data such as XML, a Database, and a Distributed Server.
3. Voice Interface Features and Challenges

The basic design principles that form the basis of a good quality web page are very different from those used to create high quality voice pages. VoiceXML applications are designed to follow a certain pattern of questions and answers in order. Moreover the presentation of transient voice content is generally event driven and chronological. These qualities impose rigid structure on end users and can make them feel lost and out of control. However, robust fault tolerance may be built into voice applications, enabling them to recover and behave gracefully, for example, in the event of unmatching or missing input. As a guideline, the user should be given lucid indications and limited valid choices regarding what input is expected at any point. Also, users may access “Help” at any point in a dialog by simply saying, "help".

To speed up the process, VoiceXML has provision for “barge in” and “mixed initiative” forms where users may give input at any point in any order in a dialog and interrupt a prompt. During any voice transaction, there are certain dialogs that are used repeatedly such as, confirmation of previous input, collection of credit card number etc. And there are many reusable components and abstraction available in VoiceXML such as Sub-dialogs, SpeechObjects, Voicelets etc. A grammar defines a set of valid expressions and inputs submitted by the user. Building grammar is the most important, arduous and crucial part of Voice User Interface (VUI) design though it may appear very simple. Underlying execution of Voice Interface, the platform should dynamically and flexibly make grammar or collection of grammar active.

VoiceXML is armed with many capabilities that allow defining scope and complexity of reusable grammar effortlessly and enable to deliver speech with the right intonation, emphasis, pitch, speed and so on.

In general, businesses will have to choose their own right way to segregate VoiceXML application between client and server which can be broken down into the following components: VoiceXML Server contents and resources, Client side Script and Application Server Components. Currently, the “Thin client” architecture is more dominant and economical. In the long run, as adequate computing resources on client devices become more common, the “Fat Client” architecture will be employed, improving system performance and scalability. Further the companies have the cost-effective option to use the voice gateway supported by voice portals, though with a quality, performance and security tradeoff.

In case of HTML Web, most frequent inputs are obtained using basic and refined controls like text box that are fundamentally “Write Anything”. There is no equivalent in Voice World due to technology constraints. Nowadays, it is the hot topic for research. One alternative is to do recording of user utterances in some audio format. But this option has quite limited and rigid scope. It does not allow simultaneous recognition of utterances and navigational control during recording. Still, recording capability can be fit for some kinds of applications like Dictation service and voice mail messaging.

VoiceXML application may be secured with authentication and access control. In voice applications, there are three approaches for authentication: standard username/password security, device ID/ PIN and Voice recognition using voiceprints. The latter form of bio-authentication has potential to significantly improve the security of applications while being convenient for users.

4. Conclusions and Future Scope

Unification of voice power and universality of telephony network with the matured web technologies has opened new vistas of voice-based solutions and an extensive new market for commerce. VoiceXML 2.0 empowers the developers with incisive tools for defining and fine-tuning the behavior of the voice interface and for grammar building. VoiceXML makes it possible for enterprises to replace legacy, proprietary IVR platforms with a unified architecture for delivering voice web services from any telephony device.

Presently, the voice world is in its incipient stage. Development of more and more voice based enterprise applications and extensive technological innovation will lead to evolution of the voice technologies to incorporate many unanticipated and application specific features. And eventually it would uncover and establish some robust design patterns for the voice solutions to capitalize on the very nature of voice interface.

5. References