COMMUNICATING WITH SAM

Leveraging Web Applications: IVR versus VoiceXML

Question:

Over the past few years our organization has placed significant emphasis on developing self-service applications for our Web site to assist in customer service. While this has been very successful in attracting new customers and increasing the number of transactions we can handle, it has not dramatically reduced the number of calls that our call centre handles to do essentially the same thing. Many long-time customers are more comfortable using the telephone to do things as they always have.

Sam Johnston

Sales and marketing have asked if there is a way in which we can leverage our Web development to better manage these calls. In the past, we have contemplated introducing an IVR front-end to our call centre, but have backed off due to the initial investment and the fact that many of the technologies are proprietary and costly to support on an ongoing basis. However, I understand there are now IP-based IVR platforms and would like to know if you think they would alleviate some of our concerns about introducing an IVR front-end.

Answer:

Long before the Web popularized the self-service model for interacting with customers, many call centres implemented IVR (interactive voice response) applications to front-end call agents. The theory was that many transactions handled by call centres were generic and common, and thus could be handled by guiding customers through a sequence of prompts using the touchtone keypad to enter numeric responses, also known as DTMF (dual tone multi-frequency). The goal was to reduce the number of low value-added calls handled by an agent by automating repetitive customer interactions, reducing the average cost per call.

While the theory was good, the practice has been less than stellar and customer acceptance has been low. Many organizations still have to augment the IVR environment with agents that are available when callers zero-out in frustration due to the cumbersome and time consuming navigation of the IVR application. The proliferation of cell and cordless phones has increased this frustration due to the keypad being integrated with the handset, unlike a traditional handset that separates the keypad from the receiver making it easy to listen and input at the same time.

While speech recognition technologies have assisted in making traditional IVR applications more user friendly, they still typically require proprietary hardware and interfaces.

This combined with complex integration requirements and limited inter-operability has made application development costly, slow and rigid, versus the fast-paced dynamic Web world.

IP-based IVR systems will address some of the issues associated with traditional IVR systems, such as lowering costs and increasing the speed of deployment by leveraging open standards and Windows-based scripting for application development, but they still do not fundamentally change how customers use the application, which is the main reason why many IVR applications have a low ROI.

While speech recognition technologies may make it easier for the IVR world to replicate your Web applications, the IVR development process will remain extensive and may still fall short of your expectations despite a high price tag, and ultimately you will still have two parallel universes – one for voice, and one to support your Web presence – which flies in the face of current wisdom if you accept convergence as an eventuality.

Based on the needs you have described, I think you would be better off looking at VoiceXML as a means to leveraging your Web presence in your call centre. Leveraging speech recognition and/or text-to-speech technologies, VoiceXML is a relatively new technology that incorporates the flexibility to create speech-enabled Web-based content or to build telephony-based call centre applications. VoiceXML is a standard markup language similar to HTML that will permit the easy extension of Internet technologies such HTTP, Web applications and media streaming to telephony applications for speech-enabled interaction. While your goal is to replicate the Web browser applications for the call centre, the outcome may be even greater, as VoiceXML, if you chose to, will enable you to permit voice transactions from your Web site for improved customer service.

In simple terms, VoiceXML provides a common language to follow when developing a speech application. These rules, also known as tags, like in HTML, denote actions for creating dialogue between the spoken voice and a speech recognition system. The VoiceXML tag would prompt the user to provide audio input, and just like a Web application, there would be defined rules for interpreting that content.

▶

The defined vocabulary and grammars that will be recognized by the system are crucial to the efficiency of application, not to mention user friendliness, as is the structure and flow of the application, similar to creating a good HTML application.
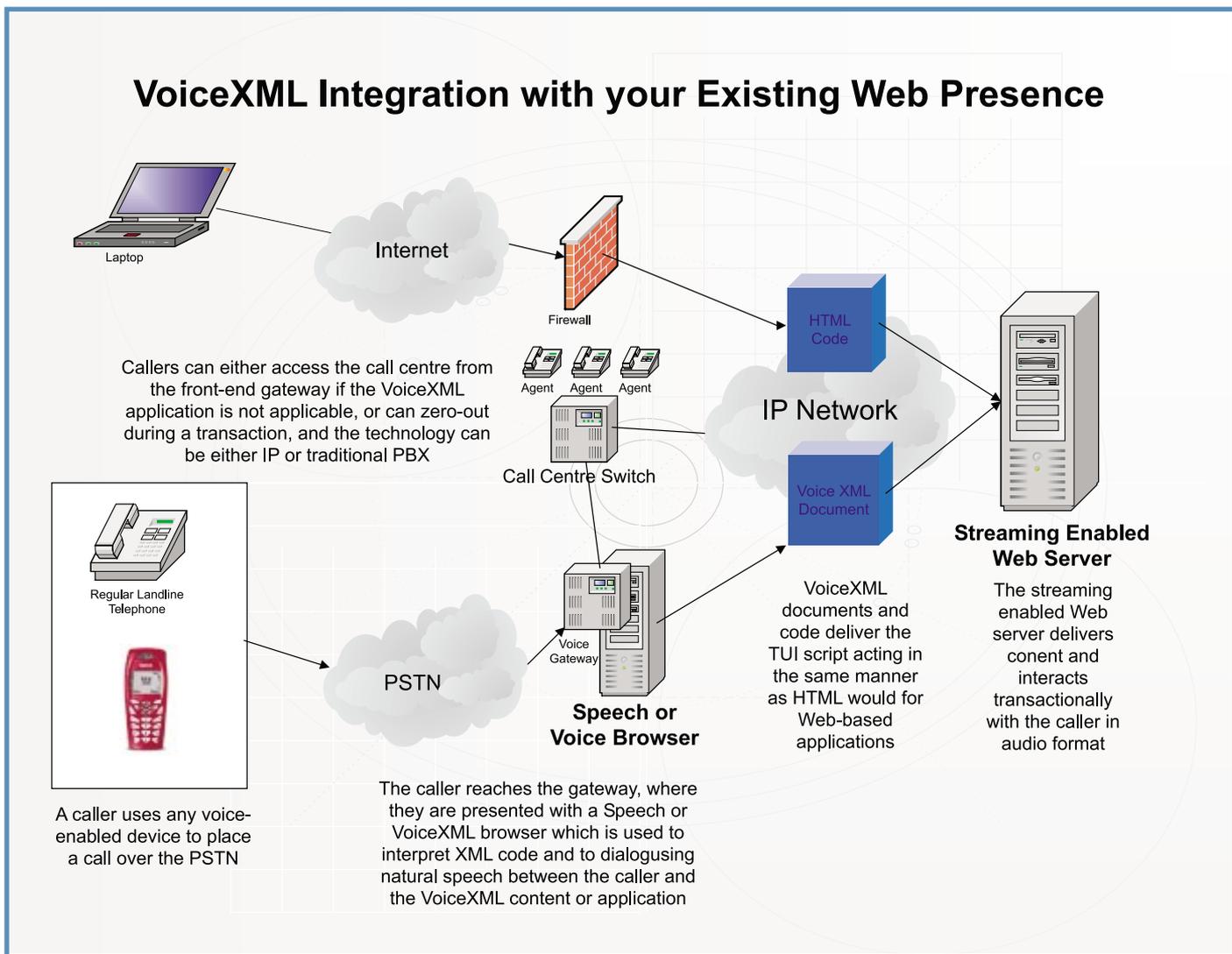
The key to VoiceXML is the interpretation and presentation of content via the voice browser, just as HTML content is presented visually to a user via a Web browser. In the end, your goal is to create an application that lets the caller speak in a natural, but structured manner, just as HTML applications prompt users to input or click in a logical and natural manner. Well developed Voice XML applications should make it easier for the user to navigate versus traditional IVR applications for several reasons:

- The elimination of the need to input numeric key pad entries to navigate will be easier on all users, and cell phone callers in particular, and this in combination with the reliance on the natural spoken voice will improve the quality of the experience making the overall transaction more consistent with interacting with a call centre agent reducing the zero-outs that can lead to increased costs per call
- The use of grammars and not just singular words will increase the navigation possible at each prompt, reducing the total transaction time, again reducing zero-outs to the call centre
- The architecture of VoiceXML will lead to easy replication of the visually presented HTML applications

- for example, text-to-speech technology means content from the Web can be presented over the phone for consistency - meaning that users who are familiar with your Web site should navigate the voice side easier due to familiarity, while receiving consistent information

Organizationally, there are many benefits beyond the improved experience the customer will have. Application flows and content developed for the Web can easily be transplanted to the VoiceXML voice browser that customers use, meaning Web investments can be leveraged for voice applications.

By converging application flows it means that data will be collected in a consistent manner regardless of how the customer interacts with you, which

VoiceXML Integration with your Existing Web Presence

Laptop

Internet

Firewall

Callers can either access the call centre from the front-end gateway if the VoiceXML application is not applicable, or can zero-out during a transaction, and the technology can be either IP or traditional PBX

Agent   Agent   Agent

HTML Code

IP Network

Call Centre Switch

Voice XML Document

Regular Landline Telephone

PSTN

Voice Gateway

Speech or Voice Browser

VoiceXML documents and code deliver the TUI script acting in the same manner as HTML would for Web-based applications

Streaming Enabled Web Server

The streaming enabled Web server delivers conent and interacts transactionally with the caller in audio format

A caller uses any voice-enabled device to place a call over the PSTN

The caller reaches the gateway, where they are presented with a Speech or VoiceXML browser which is used to interpret XML code and to dialogusing natural speech between the caller and the VoiceXML content or application

should generate efficiency in handling the transactions for important business such as order processing.

While many people have labeled VoiceXML as next generation Web-enabled IVR technology, there are key technical differences that make this revolutionary rather evolutionary. As mentioned, unlike traditional IVR platforms, VoiceXML is open standards based, using W3C CCXML for call control and W3C VoiceXML programming language, and can be hosted on Windows or Linux servers, rather than proprietary hardware devices. Circuit-based media bus of the IVR world is replaced with IP media streams, and failed API standards are replaced with the successful SIP standard. This when combined with the ability to leverage existing Web-based applications to accelerate development equates to an extremely low cost of ownership and a high ROI.

Provided that you have already invested significantly in developing Web-based applications, the incremental technology needed to add VoiceXML support may be small. Let's follow a sample call to illustrate what components you will need to add.

1. **Call Initiation from any voice device:** A caller places a call to your organization using any voice enabled device, analog or digital. The call is placed via the PSTN (Public switched telephone network) and may use the same bank of channels that service your call centre or business office, using the auto attend or a simple IVR to direct callers to the VoiceXML application. The calls will generally be terminated on a gateway server that will also transcode phones calls from G711, the PSTN standard, to an IP standard such as G729 for voice calls between an IP telephony user in your organization and an outside caller.

2. **Speech or VoiceXML Browser:** Next the caller will be presented with a Speech or VoiceXML browser, which in many cases will be integrated into the voice gateway that is used for PSTN call termination. A Speech browser provides the interactive speech dialog for the caller by interpreting and executing VoiceXML documents. The browser acts as an HTTP client to the Web server. The browser also supports interfaces to speech and telephony resources, including ASR, TTS, audio play and record functions, and collection of dual tone multi-frequency (DTMF) digits for telephone touch-tone input.

3. **VoiceXML Code:** A VoiceXML document specifies dialogs that are executed by a VoiceXML browser. A set of VoiceXML documents makes up the script for the TUI (telephone user interface) for a voice application. They perform the same function as HTML scripts do in presenting Web-based applications, and may in fact present the same content as the HTML scripts, but as an audio stream using text-to-speech software.

4. **Web Server and Content:** VoiceXML documents are retrieved from a Web server. VoiceXML browsers, on the command of natural spoken vocabulary or grammar from the caller, request VoiceXML documents from the Web server, which responds by providing static or dynamically generated VoiceXML documents. A Web server runs the application logic and may interface to external databases or application servers. The process for generating VoiceXML documents is the same as for visual Web pages and can use server-side scripting such as PHP Hypertext Preprocessor (PHP), Java Server Page (JSP), Active Server Pages (ASP), Perl, or other site-creation and management tools.

5. **Streaming Media Server:** Lastly, you will need to add media streaming capabilities to your Web server that supports the IETF Real-Time Streaming Protocol (RTSP) and Real-Time Transport Protocol (RTP) for audio playback, recording, or live audio content streaming. This will work with the text-to-speech software to present Web content via the VoiceXML browser.

Provided you have extensive in-house Web development skill, the cost of supporting VoiceXML should be considerably lower than a traditional IVR model, with the added benefit of making the content as dynamic as the content available on your Web site. Customers will enjoy using the naturally spoken words to interact with the application, meaning they actually use the application rather than needed assistance from call centre agents to generate ROI.

While we are still in the early adoption stage of VoIP (Voice over IP) and IPT (IP Telephony), it is clear that applications such as VoiceXML are the business tools that will lead to the promised-land and assist you in justifying a transformation of your voice communication needs, just as the Web transformed our data communications.

Though e-mail and the Web revolutionized how we presented data to our customers permitting businesses to collect data and interact with customers in a highly structured fashion, it could not eliminate the fundamental human desire to use the spoken word to communicate to one another. The combination of natural conversation with the structured presentation of content and collection of data means that VoiceXML may finally deliver on the IVR promise of bringing the voice and data worlds together.

TUG

Sam Johnston is a partner and Chief Technology Officer of Intesys Network Communications Ltd., providing value-added networking and e-commerce solutions to the iSeries community. He can be reached at (416) 438-0002 or via e-mail at sjohnston@intesys-ncl.com. Any TUG member wishing to submit a question to Sam can forward their typewritten material to the TUG office, or to Intesys. The deadline for our next issue is Friday April 11th, 2003.