



Colla Voce Consulting

Do You See What I Hear?

Prepared by Matt Prather, Colla Voce Consulting, 2011

Summary:

Although graphical user interfaces (GUIs) and voice user interfaces (VUIs) may provide access to the same content, to serve the same user goals, the way they go about it is — and *must* be — very different...

“The best of all possible worlds...”

Increasingly, users have come to expect that they can access the same content through a variety of means or media. And increasingly, that content has come to mean some form of self-service data and interface (for example, the TellMe™ movie listings and show times), accessible through a standard Web browser, through a WAP-based cell phone, and — most intriguingly — through a VUI, or Voice User Interface.

In an ideal world, all of these access media would work beautifully and seamlessly together, presenting the user with the same content in ways that optimized each medium. In an ideal world, each medium would deliver content in ways that still made sense to users. In an ideal world, the content could be created and structured on a single unifying principle, rather than being shoehorned into schema that depend more on the interface than the audience.

In an ideal world, lawns mow themselves....

“So groovy now, that people are finally gettin’ together...”

We, the technical community, have done a fair job of integrating standard Web and WAP (Wireless Application Protocol) interfaces, but the VUI has remained something of a “forgotten stepchild”, primarily because so many developers do not understand the particular restraints VUIS place on content. These constraints, techniques, technical requirements, and development methods can be very different from the more traditional GUIs with which most developers are familiar, and include:

- VUIs, by definition, involve the user's voice as the primary input source. This creates enormous technical challenges on the backend (the technical underpinnings that typically remain largely invisible to the user)
 - Speech recognition, noise cancellation, tuned grammars, and call flows on the back end. None of these elements apply to most traditional GUI-based Web sites. While a GUI-based Web site may include a site map, it is usually designed expressly so that users need not navigate it in a particular order, whereas VUIs must be constructed around a well-designed call flow (the planned, anticipated, and designed flow of

interaction between the VUI and the caller).

VUIs use simulated human speech to interact with the user. This presents the challenge of designing content as recorded prompts, text-to-speech, or a combination of the two. Combinations will then present further integration challenges so that they offer a seamless interface rather than a jarring "break" from one to the other. It is much easier to achieve a unified and seamless presentation within a GUI environment.

VUIs must recognize and accommodate accented speech. GUIs do not typically encounter this difficulty; a click is a click is a click, and it represents the same kind of "content" or input regardless of who inputs it within a given context.

- VUIs are invisible. To users, the VUI exists only in their minds, much as a conversation does. They can't see it, they can't refer to it, so they must remember it instead.
 - Users don't always know what to say, or what they can say. Within a GUI, the choice is usually pretty clear: enter text, click a link, scroll a window. But the VUI must usually explain its options more clearly, so that users know what is expected of them: "What can I do for you?" will work for a user acquainted with the content and the interface but a novice will usually require a follow-up prompt of, "You can ask me to open your calendar or open your mailbox. What can I do for you?"
- Likewise, VUIs are linear and sequential, by their nature. This places a higher cognitive load on users than they would encounter in a GUI: the GUI will require content structured for users to perceive, interpret, and act. The VUI, on the other hand, adds a critical step that dramatically affects the structure of the content: users must perceive, *remember*, interpret, and act. The challenge here becomes clear, in terms of content structure: chunk similar — or even identical — content so that GUI users are not required to remember each piece, while VUI users are not presented with an overwhelming number of simultaneous options.

“In the mode...”

GUI and Web content has, more and more, become structured with multimedia presentation in mind; a visual is supported by sound or a 3-D presentation or even tactile feedback and input. In contrast, a VUI is effectively a single-mode interface — sound in, sound out — and content must be designed, structured, and formatted accordingly. There is no “leeway” for information missing from one presentation format to be “made up” in a supplementary format, as can be done within a GUI.

“But how do I get home?!”

In a recent [article published online in informIT](#), the authors remind us that VUIs also present a unique challenge over GUIs: there is no true concept of “home” in most VUI environments, as there is on a typical Web site. There have been various attempts to create one, notably the Desktop of such applications as General Magic's Portico service (see Prather, [Portico Online User's Guide](#) for a discussion of this metaphor). The desktop metaphor, in particular, works well in terms of easing content-structure constraints, in that it equates well against the “desktop” metaphor of most personal computers. However, this is not at all a natural concept for a voice interface or conversation, and users are reluctant to accept or use it in this context. (See Prather & Campbell, [myTalk User Studies](#), [myTalk Design Recommendations](#))

“This aircraft is equipped with six exits: two in front, two in the rear, and two over the wings.”

Likewise, exits are not always clearly indicated in VUIs, as indicated by the number of users who continue to remark on being “lost in voicemail”. The inherently linear nature of a VUI, and the high load on cognition and memory, conspire to make the exit points, means, and trigger phrases much more obscure than they would be at similar points within a GUI.

This means that content containing an “exit strategy” must again be structured differently for GUI and VUI, even when located at a similar point within the navigation. While the GUI can afford to simply surface the content containing the exit (clearly labeled), the VUI must structure the exit content so as to explicitly draw the user’s attention and remind him/her of the availability and procedure for the exit.

XML to the rescue???

In a word, no. At least, not yet. While XML provides the content-structure framework for identifying the components of content and connecting them one to another, it still tends to do so within a bias toward a given presentation medium. That is, [XML](#) has traditionally been used to create output for a variety of visual media and formats: [HTML](#), or [WAP](#), or even print. But it has not traditionally lent itself to content creation for multiple media and is even less used for content creation in a non-visual medium.

Some thought the problem was overcome with the advent of [Voice XML](#), typically used now by most VUI developers. VXML works reasonably well within the context of a strictly-VUI environment (although it still falls down when creating a natural-language environment), but VXML has been so tightly integrated and designed for the VUI environment that it cannot then be easily used as a framework for content that will also be used in a GUI environment.

Not only is this not a two-way street, then, it’s a NO-way street. Content developed in traditional XML cannot simply be moved over to a voice interface, and content designed for Voice XML cannot simply be “poured into” simple XML for GUI presentation.

“Whaddaya wanna do? I dunno, whadda you wanna do?”

So is there any chance of truly having these two types of interface media meet in the middle and sync up, presenting the same content effectively and symmetrically? Absolutely — but the answer lies not in the tool used (XML vs. VXML), but in the awareness of each medium’s inherent constraints, and the need to carefully design to accommodate each.

VUIs can offer much of the same conventions as Web interfaces, and they can be designed and created to effectively create and mirror users’ mental models. But we must always keep in mind that a VUI is both more and less than a “talking Web page”, and structure the content accordingly.

That’s what we all “wanna do”.

References

Voice Application Development with VoiceXML by Rick Beasley, Veta Bonnewell, Mike Farley, John O'Reilly, and Leon Squire (Sams, 2001, ISBN), quoted in InformIT, 10 May 2002

myTalk User Studies / Results, by Matt Prather & Gweneth Campbell, Ph.D., prepared for General Magic, Inc., March 2001

Webley CommuniKate White Paper, Webley, Inc., 2003

Portico Online User's Guide, by Matt Prather, prepared for General Magic, Inc., 2000

Nuance Communications SSML White Papers

ScanSoft White Papers

Stanford University: "Using Design to Increase Disclosure",
<http://www.stanford.edu/~nass/comm369/pdf/Disclosure.pdf>

Turning GUIs into VUIs, by Bill Byner, VoiceXML Review, Volume I Issue 6, May 2001
http://www.voicexmlreview.org/Jun2001/features/dialog_design.html

About the author:

V. M. Prather is an independent consultant and designer, working in the San Francisco Bay area. As a technical communications professional of nearly 20 years' standing, He has worked as a technical writer, editor, designer and usability professional for such companies as Microsoft, Borland, Oracle, and General Magic. Mr. Prather specializes in integrating GUI and VUI-based interfaces to Web content, and is still reeling from having recently attended his high school reunion and wondering how all those *other* people got so old...

Contact: matt@collavoce.net