SHOPPING CART
LOG IN
MEDIA CENTER
MAILING LIST
CONTACT US
HOME

Magazine
New Customer
Management Insight
Digital Publication
Call Center Magazine
Archives
FREE Webcasts
Podcasts
Vendor Guide
Multimedia Content
Online Advice
eNewsletter Alerts
ICMI Forum
ICMI Jobs
Training
Consulting
Newsletters
Publications/Tools
Industry Research
QUEUETIPS
Conferences
ICMI Global Report
Call Center Knowledge
Online
Join ICMI
Members Only
Vendor Guide
Industry Information
About ICMI
Download the ICMI
Catalog
-

**TechEncyclopedia**

Define It

# Multimodal Toolkits: Design, In - Speech Tags and ML, Out

**Want to write mobile data applications you can talk to? While 2.5 and 3G wireless rolls out and proves itself, check out these multimodal application development tools.**

**By Ellen Muraskin**

**01/06/2003, 10:39 AM ET**

Last month, we investigated the first steps on the long march to multimodal wireless applications. We focused primarily on Kirusa (Berkeley Heights, NJ - 908-464-4467, www.kirusa.com), which makes a runtime platform. Since then, we've had a chance to look at two betas of development tools: Microsoft's, carrying the banner of SALT-based speech tags, and IBM's, carrying the banner of VoiceXML preservation and combination with XHTML. So far, SALT competes with VoiceXML for voice-only applications, so we looked up VoiceXML IDEs as well.

## Microsoft's .NET Speech SDK

The second beta of Microsoft's heavily promoted .NET Speech SDK was released at SpeechTEK at the end of October. The tool comes with a light-duty, desktop version of Microsoft's speech recognition and TTS, and works with a speech add-in to Internet Explorer, so that developers can test their multimodal applications in a browser, on standard PCs equipped with microphone and speakers.

The browser-equipped standard desktop PC, certainly, is going to be an important platform for delivering certain kinds of multimodal application. But the real thrust, of course, is towards mobility: the SDK and browser plug-ins run nicely on 802.11-equipped standard laptops and on HP/Microsoft's new TabletPC (winner of a 2003 Product of the Year Award, see article here), which holds out the promise of freely-mobile, stylus-and-voice interaction. The components

▶ UTILITIES

🖶 print this article
✉ e-mail this article

▶ RELATED LINKS

Cicero Broadens Support of Web Services and SOA

Executive Interview: Intervoices Ken Goldberg

Speech Makes Inroads As A Service

Q&A: Authentication Technology for Call Center Security

Why Mobility Will Change Telephony

Are We Finally Ready For Remote Agents?

Products of the Year: These Are the Sharpest Knives in the Drawer

Aspect and Microsoft Create Speech Application

**Free CallCenter Insider Newsletter**

**Your Email Address**

**Optional Areas of Interest**

☐ International New s
☐ Advice/Tips
☐ Technology
☐ Agent Development
☐ IVR

aren't yet ported to PocketPC OS, so totally mobile SALTed multimodality still waits in the wings.

Meanwhile, however, the toolset, requiring Visual Studio .NET, Windows 2000 Profes-sional SP3 or XP SP1, the IIS web server, and headset, will also produce SALTed voice-only telephony applications.

These speech-enabled IVRs will execute on Microsoft's.NET Speech platform (see last month's story), now in limited partner release with an eye toward joint development for the major verticals - travel, financial, and call center hosting. Since Microsoft's SALT is promised to conform to eventual W3C standardization, the pages that the SDK generates should also run on other multimodal platforms to come.

The two-CD toolset's documentation describes a two-phase application-building process. First, you design overall web page structure and call flow, create or import ASP.NET input controls, database access code, and other standard application elements. Then you define a grammar, create prompts, and finally use the Speech Controls to integrate the grammars and prompts into the web application.

The IDE lives in Visual Studio, but in its latest .NET incarnation. This fits the new multimodal functionality neatly into Microsoft's distributed-computing vision of the future, where discrete Web Services can be called across various points in a network and directed via XML. The toolkit integrates with Microsoft's ASP.NET programming environment, so - confirms James Mastan, marketing manager - any .NET-compatible scripting language, whether Jscript, Visual Basic.NET, or C# or Java, will be able to use the controls.

The toolkit adds a set of speech ASP.NET, drag-and-drop controls to the Visual Studio palette. Some of the controls encapsulate common application/end-user interactions at a fairly high level of abstraction: obtaining phone numbers, currencies, dates, yes/no, and social security number. Others give lower-level, less "cooked" access to input (e.g., get-DTMF), or let the app communicate with infrastructure (e.g., make-call). "Semantic Maps" clearly define what verbal information the application requires from the user, and where responses should be stored. Typically, recognized data is inserted into a form element on-screen, so that the same forms-handling code that manages normal browser interactions can be driven by speech input.

Marcin Krieger, .NET program manager, showed us how to add a QA (Question and Answer) speech control to the sample flight reservation app. A right click on the dropped control opens up a property builder for attaching stored grammars to that recognition event. He goes on to map the recognized word to the semantic item, completing the input connection.

Grammars can be built from scratch or modified in the Grammar Editor, a

separate window. This appears to be a fairly sophisticated tool for building mixed-initiative grammars; that is, letting the speaker fill in one or many fields in the course of one reply, and figuring out which questions remain to be answered.

The Grammar Editor shows varied paths for recognition grammars. For example, the "preamble" specified in the sample flight-booking application lets the user say "I'll have a:" or "I want a" or "Give me a." Words like "vegetarian" or "kosher" or "low-salt," say, in a meal request, are tagged to be spotted, and are given properties that bind them to the control and to the semantic map in turn. Adding prompts in TTS is similarly a property form-filling process, as is activating recognition events with start events.

The SDK renders the drag-and-drop designer into ASP.NET. "It's our markup that is then turned into SALT," says Kreiger. "But you can hand-tweak them and write SALT directly into the application without any controls," he adds.

A set of grammar libraries is included, written in W3C-approved SRGS format. A prompt management tool is especially interesting, in that it makes use of recent speech advances in automatic prosody tuning. This automatically blurs, if not exactly hides, the seams between the parts of a sentence that are prerecorded speech and those that are TTS-generated.

Mastan says that he would point developers to the voice ASP Hey Anita (Los Angeles, CA - 323- 692-1500, www.heyanita.com), which has incorporated a SALT browser, to port and test their SALT applications on a full telephony platform.

**IBM's X+V Toolkit**

IBM's (Armonk, NY - 914-499-1900, www.ibm.com) Multimodal X+V toolkit, also in beta, is firmly rooted in its WebSphere web application server strategy. The XHTML + VoiceXML multimodal markup was submitted to the W3C by IBM, Opera, and Motorola last year, and IBM's SDK is basically a modified version of its VoiceXML toolkit, adding Eclipse-based plug-ins to web developers' existing WebSphere Studio IDE.

Like the Voice toolkit, it's basically a text editor, with automatic indentation, some coaching, the usual canned dialog components to recognize common things such as yes/no, social security number, and the like, and debugging help that knows the rules of both MLs.

IBM's take on multimodal applications preserves the existing intellectual, hardware, and software investments in VoiceXML and marries that to XHTML and its own intermediate piece, which manages application states regardless of interaction medium. IBM calls that piece Interaction Manager.