opusresearch

Last Man Standing: The Future of Independent Application Development Tools

A Conversational Access Technologies Advisory
June 2006

Avery Glasser
Analyst

With Cisco's recent purchase of Audium, the number of independent, voice-application development tool providers in the market has been reduced to one: Vicorp. As the independent tool market dissolves, the focus shifts towards providers of niche functionality and service differentiation that hasn't been addressed by the classic tool vendors. Being the last man standing, Vicorp is now positioned to build OEM relationships with Nortel, Intervoice and Genesys – the three voice self-service platform vendors with multiple platforms and without an appropriate developer tool strategy – or even become an acquisition target. As the last of the independent tool providers, Vicorp still has plenty of value.

The Independent Tool Market Is Over

The apparent dissolution of the independent, voice-application tool market isn't a mark of failure for the business model. In fact, it's the sign of a maturing market. Platform vendors are determined to move up the stack due to increased pressure to bring immediate value to application implementers.

It's no surprise that the companies in this space have been acquired by platform vendors or have shifted their focus to bind their tools to a specific suite of applications. For most of the erstwhile independent tool providers, focusing on vertical applications has become the path to increased revenue.

For most of the erstwhile independent tool providers, focusing on vertical applications has become the path to increased revenue.

To this point, Apptera, Fluency and TuVox have focused on creating various levels of "prepackaged applications and application elements," using their tools to increase speed to market and manage bound applications.

Audium, one of the longest standing independent tool providers, was recently acquired by Cisco to fill out its product offerings. VoiceObjects – which had always defined itself as an application server, not a tool – has adopted a strategy of embedding its solution into larger offerings through partnerships and OEM relationships.

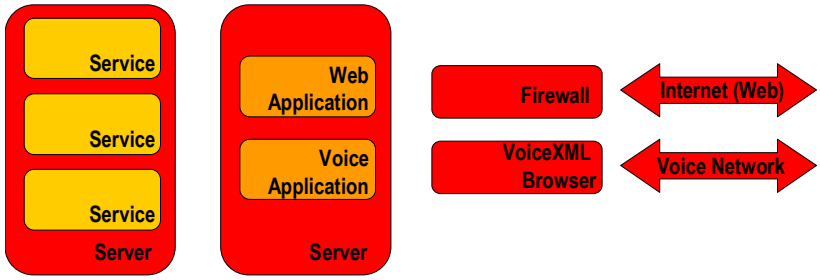
From the initial list of independent voice application tool providers, there is only one vendor of note left: Vicorp.

Development Tools in an IT World

Historically, voice applications were built using a monolithic architecture: coded in proprietary languages, compiled into executables and deployed onto the IVR that processed the call. With the introduction of VoiceXML, voice applications shifted to become more Web-like: applications coded using standards-based languages, deployed onto industry standard application servers and interpreted on the fly by a voice browser.

To understand the reduced need for independent tools in this new standards-based voice ecosystem, one needs to understand the parallels between VoiceXML voice self-service platforms and their Web analogues.

Figure 1: Voice and Web Logical Architectures



From a logical perspective, voice applications are no longer monolithic applications with self-contained logic. Now, voice applications, like their Web cousins, are assembled by connecting services together into applications which then are presented to a browser (VoiceXML or HTML). [See Figure 1]

From a logical perspective, voice applications are no longer monolithic applications with self-contained logic.

The services are generally accessed using standard methods such as Java classes, Windows DLLs and Web Services/.NET APIS. Applications take these services, apply business logic and then provide a presentation interface, rendering the application as HTML for the Web or VoiceXML for telephony services.

The core decision of what tool is used to develop applications (and the services utilized by the application) is predicated on the application server that houses the Web-based, self-service application and the tool provided by the application-server vendor. For example:

Application Server	Technology	Vendor Supplied Dev Tool
Microsoft IIS	.NET	MS Visual Studio
IBM WebSphere	Java	IBM Websphere Studio (Eclipse)
BEA WebLogic	Java	Eclipse w/WebLogic Plugin
Oracle AS	Java	Oracle JDeveloper
Sun Java AS	Java	Sun NetBeans
RedHat JBoss	Java	JBoss Eclipse IDE
SAP NetWeaver	Java	NetWeaver Studio (Eclipse)

Java-centric enterprises also have the option to use the generic open-source Eclipse Integrated Development Environment (IDE), regardless of the vendor-supplied toolset. However, using a version of Eclipse without platform-specific enhancements (most commonly plug-ins) makes application deployment and management more difficult.

To this point, even Oracle and Sun, who provide non-Eclipse based IDEs by default, still provide plug-ins for Eclipse, essentially creating a vendor-supplied (or at least "vendor-blessed") tool.

For enterprises using the open-source Apache Tomcat application server, Eclipse is the most common development tool. The primary reason an alternative application development tool is brought in is when a proprietary niche technology is introduced into a self-service application, such as Macromedia Flash and Shockwave. Of course, Java-centric enterprises also have the option to use the generic open-source Eclipse (IDE).

For enterprises, the key to standardize Web application development is using one of these vendor-supplied tools. Traditionally, the only time an application server tool isn't used is when interfacing with a proprietary technology.

For enterprises, the key to standardize Web application development is using vendor-supplied tools.

Once an application has been completed, the user interface is grafted on (traditionally using the vendor-supplied tool) and the Web application is ready for use. It is important to note that, when developing of Web applications, it's rare to find enterprises using anything but tools provided by their application server vendor.

Leveraging IT into Voice Application Development

Now that voice applications are starting to resemble distributed Web applications, the same model still stands: The tools supplied (and supported) by the application server vendor are the ones used for development.

As there are still artificial (and typically political) separations between the voice and IT/Web organizations, voice self-service platforms traditionally portray themselves as application servers, even if they are using JBoss, Microsoft IIS or Apache Tomcat as an underlying server technology.

Following the Web model, these voice self-service platforms act as "voice application servers." The expected tools to be used would be those provided by the platform vendor.

Voice Application Server	**Vendor Supplied Dev Tool**
Cisco CVP	Cisco (nee Audium) (Eclipse)
Genesys GVP	Genesys Studio (Proprietary)
Genesys/VoiceGenie	NONE - 3rd Party Certified
Avaya IR/VP	Dialog Designer (Eclipse)

Microsoft MSS	MS Visual Studio (Proprietary)
IBM WebServer Voice	WebSphere Voice (Eclipse)
Nortel Web-Centric Self Service	WVADS (Eclipse Plug In)
Nortel VPS/MPS (ex. Periphonics)	MPS Studio (Proprietary)
	Vicorp xMP – 3rd Party Certified (Eclipse)
Intervoice	Media Exchange Studio (Proprietary)
	InVision Studio (Proprietary)
	Edify Voice Studio (Proprietary)
Voxeo	Vicorp xMP – 3rd Party Certified (Eclipse)
Envox	Envox 6 VoiceXML Studio (Proprietary)

The proprietary technologies used in the voice arena today focus on prepackaged applications, and include solutions and associated tools offered by Apptera, Fluency and TuVox.

Following the Web model, a mature voice market doesn't necessarily support independent IDEs. Enterprises tend to use the tools provided by the vendor except in specific cases when the vendor-supplied tool wasn't comprehensive enough – or in the case of VoiceGenie, not supplied at all. The promise of an independent tool vendor mitigating risk regarding the selected voice self-service platform or to assist multi-voice, self-service vendor enterprises doesn't apply.

Also following the Web model, it's expected that the days of proprietary IDEs – ones not based on Eclipse or Microsoft Visual Studio – are numbered.

Harvesting the Low Hanging Fruit

Based on the reality that all non-Microsoft tools are migrating towards Eclipse (either as the primary or a supported alternative), the voice self-service providers most immediately in need of a development tool overhaul are Genesys (GVP and VoiceGenie), Nortel (MPS/VPS product line), Intervoice (Edify and Intervoice Media Exchange product lines) and Envox. Sitting in the middle of this time-sensitive "Build, Partner (OEM) or Buy" decision is the remaining independent voice application development environment vendor, Eclipse-based Vicorp.

Vicorp's decision to support both Eclipse and an OEM-partnership model is fortuitous. Without a public Eclipse strategy, any or all of four platform vendors mentioned above could benefit from using the Vicorp xMP at the core of their application development tool strategies.

Furthermore, any of these platform vendors could also benefit from acquiring Vicorp, requiring the remaining three to 'fend for themselves' and build an Eclipse strategy from scratch. Because Nortel, Genesys and Intervoice currently suffer from the post-acquisition syndrome of having to support multiple platforms, Vicorp could help reduce overtaxed engineering resources by supporting all the platforms listed above.

For the last independent vendor standing, the prospects have dramatically increased.

Tomorrow's Development Tool Landscape

As IT and Telecom environments merge inside enterprises, the need for a specific Web self-service IDE and a voice self-service IDE will start to disappear. Microsoft, with Visual Studio, already offers a single development environment that can author Web and voice self-service applications.

To leverage the dominance of the Eclipse tools in the Web self-service space, the voice self-service platform vendors need to start looking at their Eclipse-based development environments and re-architecting their platform-specific functions as plug-ins that can be added into the Eclipse offerings from BEA, IBM and other application server vendors.

Again, this model follows what can be considered to be the first axiom of development tools: Application tools are tied to application servers, not [Web or voice] browsers.

First axiom of development tools: Application tools are tied to application servers, not [Web or voice] browsers.

Tools Need to Alleviate Frustration

The myth that a development tool is required for developing VoiceXML scripts is as unsupportable as it would be in the Web world. In the Java and .NET world, developing an application can be easily done with a text editor – but IDEs that hook into version control, compiling and debugging tools are key to supporting ongoing development.

This leads to the second axiom of tools: The necessity for a tool is based on resource availability, application complexity, technology obscurity and professional services cost.

To this point, finding Web developers who can learn how to write VoiceXML is simple: markup languages are straightforward for any programming professional. However,

Second axiom of tools: The necessity for a tool is based on resource availability, application complexity, technology obscurity and professional services cost.

developing tools that make it easier to build effective voice user interfaces as well as grammars – especially for natural language applications – are of critical importance to the industry and application developers alike.

Microsoft, which does not have a designated, inhouse voice professional services arm, has a distinct need for building tools that will allow the millions of Microsoft Visual Studio certified developers to start building applications.

Leveraging technologies acquired from the buyout of Unveil, Microsoft promises to release tools for natural language grammar development, testing and analytics which will be available for users of its Microsoft Speech Server 2007 customers as well as VoiceXML users in general. Also working on similar tools for developing natural, dynamic grammars is Sympalog, a German firm developing a suite of tools to handle just these tasks.

Third axiom for tools: When a function is obscure or complex, or when resources are limited and expensive, even a standards-based technology appears to be proprietary.

However, based on release schedules and pre-release briefings, Vicorp's Eclipse based "Narrator" product is in the leadership position today. Scheduled for an end of year release, this product allows self-service application designers to enter conversations into the system. These conversations are compared against embedded best practices and recommended dialogs are created, including recording lists and grammars. This "voice user interface (VUI) package" can then be grafted onto a self-service application by a developer much easier than developing from scratch.

Because the resources for appropriate designers are scarce (and expensive), independent development tools in this space are much better tolerated.

This leads to the third and final axiom for tools: When a function is obscure or complex, or when resources are limited and expensive, even a standards-based technology appears to be proprietary. In this case, Vicorp is well-positioned – not necessarily as an independent tool designer, but by acting like a proprietary technology and tool provider.

Of course, tools don't obviate the need for VUI designers. By definition, they reduce the need for specialized resources for simpler tasks – and, as the tools become more polished, the need for designers is reduced even further. For extremely complex or sensitive applications, nothing will remove the need for VUI and grammar specialists.

However, for the majority of self-service applications out there, developers will be armed with the resources necessary to create appropriate customer self-service solutions for voice as they are with the Web.
