Whitepaper:

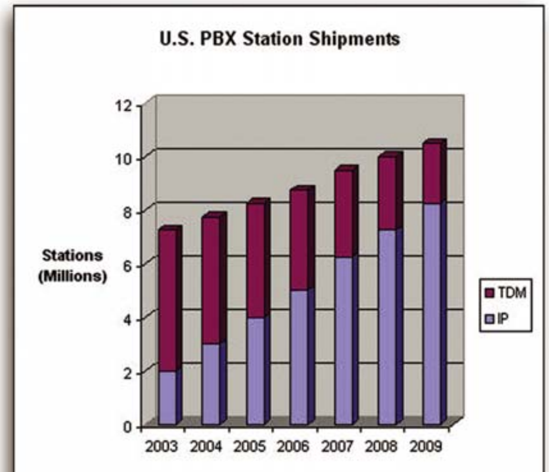
Common Pitfalls and Successful Strategies for VoIP Applications That Can Give Your Business a Competitive Advantage

Converged applications demand a new approach. What can you learn from the early adopters?

Author: **Louis Marascio**

Louis Marascio is CTO of Metreos Corporation, a leading VoIP application environment provider that helps enterprises rapidly develop, reliably execute and centrally manage applications that integrate voice communications with existing enterprise applications, automating and streamlining business processes for competitive advantage.

As expected, the enterprise market has embraced IP communications for toll bypass and cost reduction, with IP end station shipments surpassing digital shipments in 2005. VoIP infrastructure adoption is no longer in question, but the far greater benefits of converged communications and streamlined business processes remain largely unrealized. Enterprises are now demanding applications that leverage their IP communications investment by integrating voice with enterprise applications and content – not just in the call center, but throughout the entire business.



Source: Business Communications Review, Jan 2005

Early adopters purchased and developed converged applications with mixed results typical of such early efforts. The benefits of new converged VoIP applications are exciting and promising. For example, broadly targeted applications allow mobile employees to securely use cellular or home phones to make calls through the corporate IP PBX, or provide single number reach to mobile employees' office, cell and home phones, or eliminate the need for VPN routers when placing IP phones in employee homes. More industry-specific applications can streamline or completely transform a company's core business processes. (For more examples of what early adopters are doing, see the Metreos website.) However, the cost, complexity and risk of buying or building these new applications have proven to be significant.

Got Dial Tone?

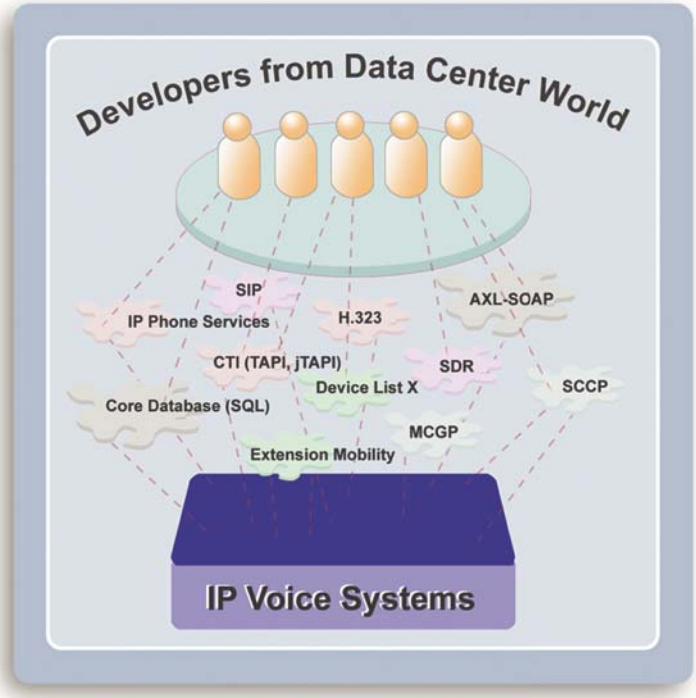
In the early days of VoIP adoption, enterprises learned a great deal about the planning, implementation, and ongoing network infrastructure management required to ensure the reliability and QoS of basic voice service delivery. Dial tone reliability is fundamental to business continuity. Many lessons continue to be learned as VoIP users discover that modifications to routers and switches, updates to server configurations, and changes to any part of the distributed networking infrastructure can now impact the availability and performance of voice communications.

When you add VoIP applications to the infrastructure issues, reliability concerns become even more serious. Early adopters have encountered a number of problems. For example, many IP telephony protocol implementations are not yet mature, and while many VoIP platform vendors utilize standards such as H.323, MGCP, or SIP, they often force those standards to work within their environments by adding proprietary extensions. Additionally, users have found that many APIs and protocols are very unforgiving, and errors in their use can result in crashed IP phones, or worse, adversely affect call processing on the switch. Even to the extent that the protocols work correctly, improper or inexperienced use of the protocols can easily impact voice availability and performance.

The core problem underlying these kinds of issues is that IP PBXs expose development interfaces, but they do not provide the robust capabilities required in an application environment. And all these new applications are running directly against the PBX. There is no middleware to buffer and protect the IP PBX from any variety of ad hoc application access and use. There is no way to ensure that the applications will not impact basic dial tone and core call processing.

Middleware All Over Again

There really is nothing new under the sun. In addition to the lack of a distinct separation between call control logic and core call routing, early voice application developers have encountered many of the same problems that most recently plagued the early developers of Web applications before platforms such as J2EE, LAMP, and .NET became available and reasonably robust.



1. The Complexity of How – Developers must worry too much about how to do things rather than focusing on what they want to do. The wide variety of telephony protocols and vendor-specific APIs presents significant complexity, and developers tasked with creating innovative converged applications are struggling with the learning curve to successfully use these protocols. There is a need for protocol abstraction, similar to the way that application servers abstracted complexity in the Web environment.

2. Lack of Experience – As VoIP causes responsibility for telecom to migrate into the IT data center organization, IT developers are being asked to develop voice applications with little or no experience in the telephony domain.

3. Building From Scratch – Early developers of voice applications have also found that they must develop all the functions they need from scratch. There is no comprehensive library of pre-built and pre-tested functions available to pick and rapidly integrate into a coherent solution.

4. Unique Voice Requirements – Like each new application architecture before it, voice presents unique new requirements. IP PBX's like Cisco's CallManager include basic support for voice processing, but they do not expose a robust media processing engine capable of sophisticated conferencing and prompt handling, text-to-speech, speech recognition, and other more advanced voice processing capabilities. Developers have repeatedly run into roadblocks as they try to deliver the solutions demanded by the business.

5. No Visual IDE – Developers today have little in the way of application development tools. They must download and study protocol specifications and experiment with the use of the protocols to see what capabilities are available and if and how they work. Those tools that do exist primarily focus on solving the problem within the limited scope of inbound IVR definition. There certainly has not been a good visual integrated development environment that can ease voice application development across the entire problem domain – from call control and media processing to IP phone display manipulation and integration with Web services.

6. No Automated Test Tools – Once applications are developed, they must be tested, and again there have not been good tools available to automate this effort. For example, if a developer wants to see what is going to show up on the IP phone display when the program runs, he or she must look at an actual phone rather than having an integrated QA tool that can be used for functional or automated regression testing.

Manageability – An Afterthought

Beyond application development and test, there is a need to take a more holistic view of application lifecycle management that extends to operations management and support, and as has been the case in most new technology markets, management has been an afterthought.

Early users of IP voice applications have found that once new applications are developed or purchased, there is still a lot more ground to cover. Initial deployment of the new applications can be tricky and time-consuming, and ongoing configuration and management of upgrades is difficult without appropriate automated management tools.

In addition, enterprises have found that once they get past the first and perhaps second deployed application, they struggle with the task of managing scalability, capacity, and performance in a structured or predictable way. There has been no application server-like container that can be used to deliver standardized scaling and operations management for a set of voice applications.

Legacy Call Center Solutions

It's clear from a variety of early adopters' experience that there are significant challenges in the development of converged voice and data applications, but it seems there should be better solutions. After all, pulling voice and enterprise applications and data together is not a new problem, is it?

Voice application development is not new, but developing converged IP voice applications that can transform business processes and empower employees throughout the enterprise is new and does create new requirements that must be addressed by the vendor community in order to enable enterprises and integrators to leverage the promise of converged voice and data networks.

Computer telephony integration (CTI) represented the first significant attempt to get voice and data working together. Technology vendors used CTI to craft interactive voice response (IVR) and intelligent call routing (ICR) tools that were purpose-built for developing call center solutions – i.e. customer facing applications for handling in-bound calls. These solutions have proven to be problematic in two primary ways for use in developing more broadly targeted converged applications.

First, CTI itself in the form of the TAPI and JTAPI protocols is problematic. With a great deal of care and expertise, CTI can scale to accommodate a call center, but using CTI for a broadcast voice application to reach 10,000 employees or to enable secure, remote soft phone access for several thousand insurance agents is not feasible. Other lighter weight protocols make much more sense, and perhaps the protocol of choice varies among vendors and the version of each of their IP PBX products.

Second, because the legacy call center tools were purpose-built for solving call center problems, they offer functionality that is (appropriately) focused on call center requirements rather than the more generalized architecture and capability required to support applications used by all employees throughout an enterprise. For example, these tools are primarily crafted around developing IVR trees to handle inbound customer calls. They expose limited capabilities for additional media control (such as conferencing) and aren't flexible enough to develop more general-purpose voice applications (e.g. receive an SNMP trap, determine the owner of the troubled system from LDAP, initiate outbound calls to troubleshooters, conference all parties, and use text-to-speech to read error information). Finally, the majority of these tools are wholly dependent on the call center infrastructure to operate.

A Real-World Example

So far we've been exploring the challenges encountered in general by a variety of different enterprises that have taken the lead in using converged voice and data applications. Let's illustrate these problems more specifically by examining one such enterprise.

Lehman Brothers is known as an innovative leader in the use of technology for competitive advantage. Lehman standardized on Cisco's IP telephony product line to gain the tactical cost advantages associated with the infrastructure migration, with an eye toward the more strategic gains to be had by leveraging the technology to impact their core business processes. Like others Lehman established responsibility for its IP telephony infrastructure with a voice networking organization placed side-by-side with the existing data networking organization in the IT department.

As Lehman began the development of converged applications, one of the first business processes Lehman decided to revise was the process by which Lehman analysts informed their customers of important information on a day-to-day basis. The process worked as follows. An analyst would put together a list of the customers affected by the information. He or she would then look up and dial the phone number of each customer. If the customer answered, the analyst would read the information to the customer on the phone. If the customer's voice mail answered, the analyst would read the information to the customer's voice mail system. Then the analyst would look up the next customer's phone number, and so on. Analyst time is a valuable and expensive resource, and Lehman decided it could streamline this process, saving a considerable amount of its analysts' time each day.

To do this Lehman set out to build a converged application. One component was to be a Web-based application using IBM's WebSphere Application Server that would automate the generation of the customer lists. Then the application would auto-dial the next customer's phone number on the list through the Cisco CallManager IP PBX from the analyst IP phone. If the customer answered, the analyst would read the information to the customer as before. If the customer's voice mail answered, the application would automate playback of the information as a recorded message and immediately move on to auto-dial the next customer number, freeing the analyst from leaving recorded messages. The application could also record calls and store them in a database if needed.

The Real-World Problems

Lehman assigned a developer from the IT data center group to develop the converged application using the C++ language in Microsoft Visual Studio and the TAPI protocol. The Lehman developer spent months writing this application. He ran into one roadblock after another, from trying to make sense of the telephony protocols that could be used to the need to license and integrate a media processing engine. After a great deal of time and frustration, he finished the application.

From a strictly functional perspective, the QuickDial application did what it was intended to do, but it was highly problematic in almost every other respect. For example, the application was not sufficiently stable,so it would frequently crash. And when it crashed, it would sometimes bring down the CTI Manager component of Cisco CallManager, de-stabilizing the IP PBX and rendering CTI Manager unavailable.

The Lehman developer tried repeatedly to fix these problems, but stability remained such a serious problem that ultimately Lehman was forced to deploy and maintain a separate CallManager cluster just to host and isolate the QuickDial application. This was a problem for Lehman both due to the cost as well as the time and frustration involved in maintaining the extra CallManager cluster (e.g. applying patches to both clusters, etc.).

(continued...)

In addition to reliability and stability problems, Lehman also found that the application was not sufficiently scalable due to the use of the TAPI protocol and due to the application's dependence on the IP PBX to process and handle the media streams associated with the application.

Things went from bad to worse when responsibility for maintaining QuickDial was transferred to a new developer who found it nearly impossible to understand and update the application.

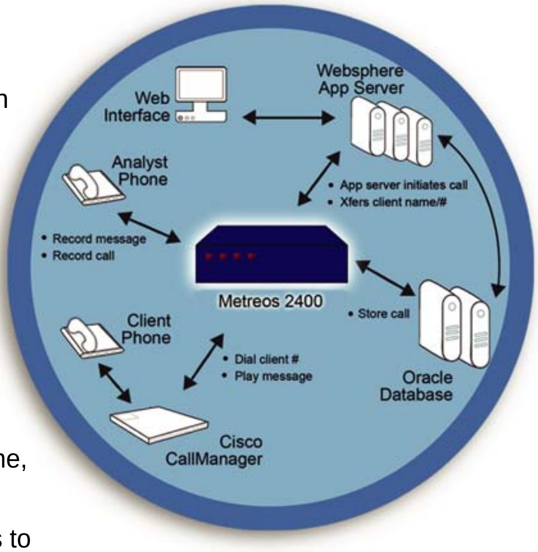
The problem was two-fold. First, this developer had little knowledge of TAPI and found it difficult to read and understand the use of the protocol by the application. And second, the developer had to dig through all the low-level C++ code to try to piece together the structure, logic and function of the application code. This is a common problem in any enterprise development effort; unfortunately, the addition of complex and sometimes arcane telephony technologies to the mix makes it a daunting task for even senior level software developers who are well versed in Web or database development.

These problems combined with the inherent reliability and scalability issues of the application resulted in ongoing negative experiences and frustration for the new developer, the VoIP operations staff and the business users of the application.

Because the QuickDial application required frequent updates and attention, Lehman's operations staff was also constantly spending time and effort to manually manage application deployment, upgrades, configuration and troubleshooting. This made Lehman Brothers aware of the need for something that would automate and centralize management of its new VoIP applications, especially as more applications were added to Lehman's portfolio.

The Real-World Solution

To address these issues, Lehman purchased a VoIP application environment solution from Metreos Corporation. Within 6 weeks, the new developer was able to write, test, and globally deploy the QuickDial application to Lehman Brothers' analysts, with an estimated annual value of $300,000 in cost savings. Since then, Lehman has successfully purchased or developed and deployed several other applications with an annual cost savings exceeding $1 million, including:

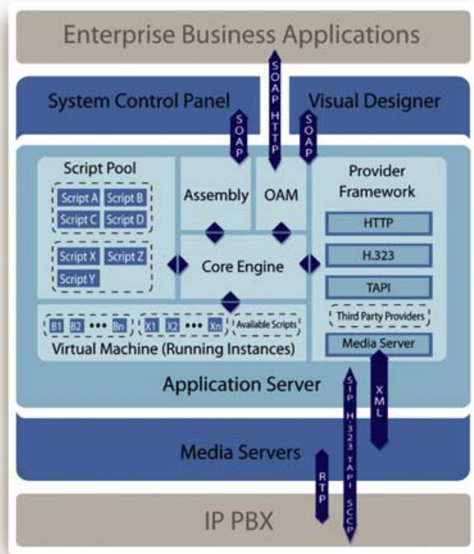


QuickDial Application Architecture

1. *Phone Proxy* - Allowed deployment of 2,500 IP phones to employee homes without the added cost and complexity of VPN routers.
2. *VoiceTunnel* - Enables employees to dial the IP PBX from a cell or home phone, authenticate themselves, and make calls as if local to the system.
3. *ActiveRelay* – A find me-follow me solution with device mobility, enabling users to have a single number and selectively move between devices without interrupting a call.
4. *Broadcast* – A Web-enabled application integrated into Lehman's home page providing easy transmission of broadcast messages throughout its worldwide offices.
5. *Jabber Instant Messaging* – Integration provides additional presence information with Lehman's IM systems to show if an employee is presently busy on the phone, tightly integrating the IP communications and IM infrastructure.
6. *Password Reset* – A replacement for an existing application providing employee self-service for password resets through a simple phone menu. Reduced help desk calls by 50%, enabling re-assignment of 3 staff to other priority tasks.

Solution Criteria

What then can we learn from history and from those who have gone before us? What requirements should an enterprise consider as it readies itself to begin the purchase or development of converged voice and enterprise applications?



Metreos Environment Architecture

1. *Telephony Protocol Abstraction* – An extensible telephony protocol abstraction framework to shield developers from protocol complexity, reduce training ramp time and enable deployment flexibility in developed applications.
2. *Core Call Routing Protection* – A run-time technology that clearly separates applications from core call routing and manages calls from applications to the PBX in order to protect the PBX from immature protocol or application defects and developer inexperience and misuse.
3. *Media Processing Engine* – A native media engine to supply ready-to-use sophisticated media processing functions commonly required by converged voice and data applications. In addition to enabling more feature-rich application development, this offloads media processing from the PBX, saving precious slots and avoiding processing spikes that can impact PBX performance.
4. *Pre-Built, Pre-Tested Functions* – A library of pre-built, pre-tested commonly required functions for call logic control and other uses to accelerate the development and testing of integrated applications.
5. *Standardized Application Container* – An application container (similar to J2EE application servers) that provides a standardized way to manage scalability, capacity, and performance of a set of converged enterprise applications.
6. *Visual IDE* – A visual integrated development environment crafted to enable the development of converged voice and data applications.
7. *Automated Test Tools* – A set of test tools to automate functional, system, load, regression, and security tests.
8. *Centralized, Automated Management*– A set of management tools to automate the deployment, configuration, upgrade, and ongoing operational management of the applications across the distributed worldwide IP telephony infrastructure.

Food For Thought

Many leading enterprises are beginning to see and benefit from the advantages of converging voice and data networks. For example, BearingPoint's consultants use a Web-based application to reserve temporary office space as they travel between various BearingPoint offices and customer locations, but there has been no way to know if they actually show up and use the space, so BearingPoint spent an inflated amount of money on real estate lease costs. BearingPoint, a leading provider of convergence strategy and implementation services (and an experienced user of such services themselves), has developed an enhancement to this application. If/when a consultant shows up to use the new reserved office space, they log into the IP phone at that location. Calls are now automatically forwarded to their new location, and a space use database is automatically updated to indicate the space was truly used. BearingPoint now leases space based on actual use data rather than inflated estimates, saving in excess of $1.5 million per year, and as an added benefit, consultants are more readily accessible to customers.

Summary

The potential for voice and data application convergence to impact conventional business processes is enormous and far-reaching. This potential over time will be limited only by the enterprise's imagination, but in the near term can also be seriously constrained by the tools available for the job. Enterprises should carefully identify their requirements with respect to full application lifecycle management of these new applications, and evaluate and consider the new tools offered by vendors in the emerging VoIP application environment market to see which tools can best enable them to be successful in achieving the potential of voice and data convergence.