

# Pipeline

Knowledge Is Power

[www.pipelinepub.com](http://www.pipelinepub.com) Volume 5, Issue 9

## Next Generation Management for Converged Services

By Rich Schmaltz and Vadim Rosenberg

The introduction of hybrid cars pushed auto-mechanics not only to learn new skills, but also to use totally new tools for the complex electronics and software responsible for the control and distribution of power. Can a mechanic use only the traditional engine diagnostics tools to analyze loss of performance?

While we are not in the auto industry, I find this example very applicable to the transforming telecommunications industry and the new requirements for service assurance. As services look more like multimedia software applications, shouldn't our management tools change, too?

In legacy environments, a service was a function of the network. User devices were, in essence, extensions of the network (phones or cable boxes plugged into the wall outlet, which connected to the distribution box, etc.). Since the network was under full control of the operator (phone or TV cable), the service was easy to manage. The traditional network fault and performance management tools took care of networks and elements from both troubleshooting and bandwidth perspectives, and element management tools made sure that all the devices were correctly provisioned and activated. Together, that was a pretty good assurance that the service was delivered. Is this still true?



Today, with voice revenues drying out and cost-sensitivity growing, operators are looking for new revenue streams and cost saving strategies. With IT standards-based Internet technology at their side, they are moving quickly towards both objectives.

For new revenues, they are rolling out next generation value added services (VAS) like m-commerce, IPTV, presence-based enhanced interaction, geo-location-based services, games, and conferencing. In today's new environment, services are essentially applications delivered to end users over pervasive channels. Can we use the traditional approach, for example, to assure that the "MyTraffic" premium service is delivered to IPTV subscribers correctly, e.g. the information is correctly delivered to their screen, and requests for zoom-in or alternate route are executed within the expected one second? Totally unrelated to the quality of the video or sound, these interactive applications differentiate IPTV from traditional TV. Even when bundled into a packaged service, they build the competitive advantage that helps operators preserve their customer base and even gain market share in new areas.

Customer Self Service (CSS), on the other hand, is more of a cost cutting measure. Customers who can easily pay their bills on-line, submit a service request, activate new service features, or chat with a representative over the built-in IM, are less likely to email or call a live agent. And of course, there is always an e-commerce and up-sell angle associated with CSS portals. Increasingly, CSS itself becomes the platform for new revenue services like IM-to-SMS, e-commerce, and Web calling. In some cases, communication services and CSS converge to the point when one is impossible without the other. A strong Web-based self-service interface is required, for example, to update the frequently-called numbers, to order a new IPTV channel, or to re-rout VoIP calls to an alternate number. These self-service features are inseparable from the service itself and are a big part of the overall user experience.



Clearly all these new services depend on software. Of course, software is not a new kid on the block for telecommunications. Intelligent network (IN) solutions, many of which are still responsible for call control and stove-pipe service delivery, have relied on software for decades for call and session control and service execution. There are big differences, however. When a network equipment manufacturer (NEM) released an IN-based service, a management component was usually built in. And since, in most cases, the software was proprietary and well integrated with the hardware, assuring high performance and availability through built-in element management was easy. Today, with the emergence of service delivery platforms (SDPs) and the convergence of network and IT, services rely on IT standards-based technologies adapted for use in telecommunications. Web and application servers, portals, service busses and orchestration engines, databases, SIP and JSLEE-based service execution environments, along with standardized network integration interfaces like Parlay(/X), are increasingly becoming responsible for next generation services. Services themselves are becoming business transactions running through these standards-based software components.

Just like in legacy and IN-driven services, network elements were responsible for service delivery, in today's IP-based environments standards-based software components are becoming new "software elements."

Perhaps now the hybrid car analogy is even clearer. If we go back to our original example with legacy phone or TV cable services, merely making sure that your network and devices operate properly was enough to guarantee good service. To provide full service assurance function for the new converged services and CSS, full visibility into both networks and applications is required. Service assurance needs to be upgraded to include application performance management (APM) as an essential component. While for enterprise IT environments and data centers, APM has been a key piece of an architecture, in the telecommunications industry it is a less familiar tool (especially in network operations and service delivery departments). So, what are the specific requirements to APM that can help operators get the most out of their SDP and CSS?

The ultimate goal of service assurance is to prevent problems or detect them before customers are impacted. From the APM standpoint, these goals can be achieved by getting visibility into all the application and infrastructure layers responsible for delivering services to the end users or partners: the presentation layer (portals), OSS/BSS and IT infrastructure, and all the way to the edge of the network where network and element management tools can take over. Monitoring user experience in real time while correlating real response times to the end-user with the events and processes happening in multiple components can dramatically improve customer satisfaction. That, in turn, increases customer retention and consequential revenues, while decreasing costs of IT firefights.

Doing all of this is not simple; however the best practices of innovative operators provide some guidance. Alltel (now Verizon, US) relies on proactive APM to manage their highly successful "My Circle" service. It allows subscribers free calling to 10 numbers that can be updated at any time only through a Web-based interface. SingTel Optus (Australia) uses APM with their Zoo SDP to manage the real-time revenue stream between subscribers and content providers. Looking at over one-hundred communication service providers world-wide who have deployed APM to manage their next generation services and self-service, we identified some key requirements for the modern day service assurance:

- **Deep diagnostics and visibility** into the execution environment in production: portals, partner gateways, Java or .NET application servers, SIP/JSLEE, Web services and SOA, all the way to network interfaces (Parlay, Parlay/X). The root cause of a problem can be found in any component, from applications and infrastructure to objects and methods. You need to be able to measure each of these components' response times, number of invocations, and generated errors. Deep visibility helps trace individual user transactions through different components and systems, and correlates them with other events. Only this will give you a complete picture of your application's health.
- **Monitor all data all the time** to catch early symptoms as they happen. Traditionally, interval probing is used to monitor networks. Networks' topology rarely changes, and most configurations are known in advance. It is possible to design sufficient probes to test the network and detect any outages. With software applications, probing is not good enough. The number of logical paths that the execution process can take is much higher, and is much more prone to changes with software updates. Also, software problems may happen under specific sets of conditions involving usage patterns, load, and other factors. With interval probing, these problems may fall between the intervals or not manifest themselves with synthetic transactions. This is why you need to monitor real user transactions, in real time. Probing still makes sense under light load,

but it will not tell how the system would behave under the real load. Also, ALL data needs to be monitored for deep analysis.

- **Management should be non-intrusive** and have little or no impact on the managed environment itself. In most cases, solutions suitable for development or quality assurance environment are not applicable in production due to their high overhead. Today there are ways to optimize metrics collection so overall additional load on the system is minimal.
- **Role-based customizable reporting** is essential to give different roles within your organization (business managers, NOC and IT administrators, or QA and software engineers) the appropriate view relevant to their function. While the source of the performance information is the same (coming from your APM solution), the usage requirements and the actions taken are very different. This relates to all the dashboards, reports, and alarms that your APM solution generates.

The bottom line: as telcos are increasingly using standards-based software for new applications and services, managing performance, availability, and the overall customer experience is becoming a new challenge. Application performance management tools, when used proactively as part of SDP and OSS/BSS architectures, can ensure high quality of service, low costs, and top customer satisfaction, ultimately making operators successful in this new environment.

Not for distribution or reproduction.