Telecommunications service providers today desperately need an infrastructure that can help them to implement new, value-added services tailored to customers needs. Efficient, fast and cost-effective deployment of such services is essential if operators are to retain and expand their customer base. Network equipment providers who can supply systems that facilitate provision of such services will enjoy great competitive advantage in the short- to mid-term future. In other words, systems that are scalable, flexible and modular, as well as reliable, will capture the market.

Convergence leads to open standards

Ten years ago, most telecommunications networks carried mainly voice traffic. As a result, switching equipment was designed primarily for voice traffic. Data traffic was carried mainly on corporate networks with little exchange between companies. Since the spread of the Internet, however, networking has changed rapidly, to the point where the legacy voice traffic infrastructure is in fact transporting more data than voice traffic. As a result, the trend is now for carriers to migrate their voice switching infrastructure to a more generic data networking infrastructure, able to carry any kind of traffic.

New generation switches support multiple types of communication, including voice circuits, IP and ATM. This is true for central office switches as well as mobile switching centers and base station subsystems. In order to have such flexibility, these switches rely heavily on computing technology. As network operators migrate their infrastructure towards such new equipment, they can seize the opportunity to choose equipment that can also be integrated vertically, to be managed as a part of a single infrastructure that stretches from high-end business support systems down to the switching level.

Integration and convergence requires use of open standards. Gradually, even the switching world is moving away from proprietary protocols and solutions to open standards. Only standards can enable the interworking of multi-vendor, multi-technology equipment and solutions that are required to create a seamless, unified network. Internet-working is essential to creating the end-to-end information flow and work flow needed by today’s telecom operators and service providers.

High availability — a primary concern

As embedded telecom developers rush to fill the transmission gap created by the Internet and convergence, they must be sure to deliver systems that can support the high availability demands of the telco industry. This means adopting many technologies and mechanisms pioneered by the telecom industry in their proprietary platforms, and then offering them in competitive commercial hardware and software platforms. The norm today in telecom services is five nines (99.999%) availability, translating into just five minutes of system downtime a year.

To meet this exceptional high-availability requirement, embedded telecom solutions based on open standards must: support such capabilities as hotswap, where an embedded system can be serviced without disruption of services; alternate pathing where the system can automatically direct disk and network operations to a predefined alternate path should a failure occur; and execute dynamic reconfiguration that allows the OS to notify the application of a change in hardware resources.

With all these requirements in mind, on the hardware side, the CompactPCI standard is emerging as the platform of choice for embedded telecom applications because of its unique combination of high reliability and cost effectiveness. Available in a dense, rugged package with excellent cooling properties in large installations, the CompactPCI boards are ideal for high-availability applications, such as telecom switches. CompactPCI also offers the cost advantages of the PC world. It includes a standard form factor that is electrically equivalent to the desktop PCI bus and supports exactly the same interface chips as those used in desktop PCs and workstations. As a result, CompactPCI enables telcos to directly leverage the inexpensive, fast silicon developed for...
the PC world with access to the latest interconnect/processing technologies.

Although the hardware platform is a critical piece in the open-standard approach, it can be argued that the operating system environment plays an equally crucial role in achieving the performance and high-availability goals required in today’s telecom applications. A viable embedded telecom platform requires a set of software components that provide a complete platform for building network infrastructure. It is only by integrating open-standard solutions throughout the entire architecture that telcos can answer the market demand for a high-availability networking architecture that enables the convergence of voice and data technologies in wireline, wireless, switching, and access networks, in a scalable, reliable carrier-grade solution.

**Real-Time OS: an enabling technology**

When it comes to the operating system, the challenge is to integrate the real-time applications typically found in telecom systems with enterprise UNIX server applications and any legacy software applications. A combination of real-time and enterprise capabilities gives telcos a way to expand service offerings on top of the traditional real-time operating systems (RTOS)-based applications to keep pace with customers’ seemingly insatiable appetite for new capabilities — all while supporting critical high-availability needs.

The real-time operating system deployed in a telecom platform is a fundamental, enabling technology. As such, great care must be taken in selecting a real-time operating system that provides specific critical capabilities. First, it should have a highly-flexible, component-based architecture that allows different services to be configured into the runtime instance of the operating system. This configuration allows the runtime instance of the RTOS to be finely tuned, according to the underlying hardware platform, the memory footprint requirements, and the application features.

A component-based design allows a very high degree of scalability. An operating system based only on the micro core executive typically requires about 10 Kilobytes of memory to run. Additional components can be added to create a tailor-made instance to reach the level of the resources available. For example, multiple-operating system personalities and APIs can be run simultaneously on a common hardware platform using such a real-time operating system, in such a way that diverse applications can communicate transparently. This feature allows quick integration of existing applications on top of the real-time operating system’s microkernel. Then the system can be extended by adding new applications that run over standard APIs to create powerful combinations of legacy applications, real-time POSIX applications, and Java applications.

Transparent distribution is another key attribute for the real-time operating system, allowing applications to be distributed across multiple machines. The capability can be used to find the location of a process (local or remote), and then identify the shortest path and quickest execution time that can be used to reach it, managing the communication in a way that makes the process location entirely transparent to the application. Implementations also should be upwardly compatible. Developers can then reuse applications on the same core operating system for a larger range of equipment. For example, an ATM router card is developed, using a small instance of the RTOS. Later another project, to develop a broadband ATM switch, can build on the routing module, reusing the code.

**Real-time and OS UNIX convergence**

Ideally, the embedded telecom platform should combine the real-time performance of the real-time operating system with the power of a UNIX operating system. This creates an environment that can sustain a unique combination of high-availability, scalability, and network management-readiness. The result is a highly integrated, ready-to-use platform on which network operators and service providers can deploy new products and value-added services rapidly and cost effectively. As such, it covers all three tiers of the network; from line cards up to back-end servers, creating a complete service delivery platform designed for wireless and wireline switching and access for Internet and voice.

Two complementary operating systems, working hand in hand, provide the numerous advantages. That is why the real-time operating system and the UNIX environment should share the same hardware, the same set of POSIX APIs, the same high-availability APIs and the same development and management frameworks. This configuration directly reduces development costs and time lags. Anything developed on one operating system can be ported to the other easily.

The sharing between the two operating systems means developers can decide exactly how much of their solution will use the UNIX operating system, and how much will be based on the real-time operating system, but use all of the same surrounding technology. Anything they develop will be completely portable from one operating system to the other. By making it easy for developers to choose any percentage split between the real-time operating system and UNIX operating system and by making all the surrounding modules and technologies converge with both, the platform enables developers and their operator-clients to begin at any point in the evolution from real-time embedded to computer-based systems. This platform also enables developers and their clients to migrate slowly to where they want to go as they replace obsolete systems and invest in new technologies.

**OS support for high availability**

It is not enough, however, to simply have a flexible, component based real-
time operating system paired with a full featured UNIX operating system. The two operating environments together also must offer critical high availability features that enable the embedded telecom platform to support a range of high-availability services.

For example, error confinement through memory protection is absolutely essential. With this capability, different applications can run in different memory address spaces, so that if one application fails, it cannot corrupt the data of the others, or the system itself. This mechanism helps to confine errors and prevent their propagation.

Fast, automatic recovery is also an important requirement. A hot-restart feature should allow automatic restart after a system failure, in the last known consistent state, without reloading code or data from disk. When a process fails, its persistent objects are preserved, and its text and data segments are reinitialized to their original context without accessing stable storage; and the process resumes at its entry point. As a result, the application can restart with little or no interruption of service. If the hot-restart functionality applies to the entire real-time operating system, not just to applications, that ensures even higher service availability.

In-service reconfiguration is another important consideration. Processes can be loaded dynamically into the real-time operating system from disk or network without first halting the system. This minimizes service downtime, allowing services to remain available while the system is modified or upgraded. For example, when the system is running in a Public Branch Exchange, new features can be added without interrupting the basic telephone service, and without reconfiguring the entire telephone network.

**An example implementation: an intelligent network**

To show the effectiveness of open-standards-based embedded telecom platforms, it helps to walk through an example. In this case, lets look at what it takes to create an Intelligent Network. Although the intelligent networking concept was initially intended for the public switched telephone network, it is now becoming an architecture of choice for delivering services to other voice networking infrastructures, such as mobile networks, SIP or H.323-based Voice over IP, and ATM.

This evolution is driven by the need to keep a consistent service creation and service logic execution environment through different voice networking technologies. The result, however, is that Service Switching Points have to accommodate different media types. These Service Switching Points are therefore commonly split into two functions: the Switching function and a Switching Control function.

The Switching function manages individual call legs, accommodating different media types, whether they are switched circuits, ATM, or IP streams. The Switching control function handles call signaling and manages the association between multiple call legs. It presents a unified call state model to the Service Control function, independent of the media type used for carrying voice.

Standardization is proceeding for formalization of this split between the switching and control functions, for example, as represented in the IETF Media Gateway Control Protocol (MGCP). This model maps quite well on the multi-tiered embedded telecom platform concept: the switching function sits in Tier one; the switching control function sits in Tier two, while the SCF, SMF, and SRF of the Intelligent Network sit in Tier three.

The advantage of the embedded approach in this particular case is that it offers consistent platform management for all the entities supporting these functions: the Service Switching Point (SSP), the Service Control Point (SCP), the Service Data Point (SDP), the Service Management System (SMS), and Intelligent Peripherals (IPs). Overall, a modular platform offers a comprehensive and consistent management framework for seamless provisioning of services, encompassing the loading of trigger detection points (TDP) on the SSP, service logic on SCPs, and service data on SCPs or SDPs.

**Ready for service**

Only by assembling a full complement of best-in-class development solutions can equipment providers hope to provide the high-availability priorities the telecom market demands. The most viable way the converged network arena can match the level of traditional availability is to carefully select a real-time operating system that is highly flexible and supports high-availability capabilities. And if this real-time operating system can be closely paired with a full-featured UNIX operating system, it creates an end-to-end solution uniquely capable of sustaining high-availability, scalability, and network management capabilities. Ω

Fred Rehhausser is considered one of the foremost experts in the embedded computer market. Industry analysts credit him with helping to launch Sun Microsystems’ Microelectronics high availability CompactPCI family, positioning Chorus Systems as the Telecom OS, establishing Motorola as the dominant VME board vendor, building the FORCE COMPUTERS SPARC-based VME business and repositioning AMPRO’s line of embedded computers as a macro-component. Rehhausser’s experience includes more than 20 years in senior or executive management positions at Sun Microsystems’ Microelectronics, Chorus Systems, FORCE COMPUTERS, and Motorola’s Computer Group. Rehhausser earned his BSEE degree from the Massachusetts Institute of Technology (MIT) and an MSE degree from the University of Pennsylvania. He is the author of numerous papers on bus architecture and bus technology and holds several patents in the computer field.

For more information, you can contact Fred at:

Sun Microsystems’ Microelectronics
901 San Antonio Road
Palo Alto, CA 94303-4900
Tel: 408-544-0357
Email: frederick.rehhausser@eng.sun.com
Web site: www.sun.com

Fred Rehhausser is considered one of the foremost experts in the embedded computer market. Industry analysts credit him with helping to launch Sun Microsystems’ Microelectronics high availability CompactPCI family, positioning Chorus Systems as the Telecom OS, establishing Motorola as the dominant VME board vendor, building the FORCE COMPUTERS SPARC-based VME business and repositioning AMPRO’s line of embedded computers as a macro-component. Rehhausser’s experience includes more than 20 years in senior or executive management positions at Sun Microsystems’ Microelectronics, Chorus Systems, FORCE COMPUTERS, and Motorola’s Computer Group. Rehhausser earned his BSEE degree from the Massachusetts Institute of Technology (MIT) and an MSE degree from the University of Pennsylvania. He is the author of numerous papers on bus architecture and bus technology and holds several patents in the computer field.

For more information, you can contact Fred at:

Sun Microsystems’ Microelectronics
901 San Antonio Road
Palo Alto, CA 94303-4900
Tel: 408-544-0357
Email: frederick.rehhausser@eng.sun.com
Web site: www.sun.com