www.pipelinepub.com
Volume 17, Issue 7

The Fundamental OSS Challenges of Yesterday, Today and Tomorrow

By: Mark Mortensen

At the turn of the century, I gave a keynote talk at an IEEE Symposium on "The OSS Millennium Challenges." In it, I outlined what I saw as the major issues facing operations support systems (OSS) for communications services providers (CSPs). Here, we look back at how we have done with these challenges: some have been met, some are still with us, and some have been superseded through a complex set of market dynamics and technological advances. So, by way of this retrospective, let us look back on what concerned us most at the turn of the century and the major challenges we are facing today and tomorrow in network operations and OSS.

Yesterday's challenges

As we transitioned into the twenty-first century, CSPs had many challenges in the network operations and OSS arenas. We were evolving the networks quickly with IP and mobile technology, competition was growing among CSPs and the new web-scale companies were beginning their rise. But to me, four key challenges in OSS stood out. Of these four key challenges, we have solved two and made good progress on the other two, even though they seemed intractable at the time.

Grandma's chair: the legacy OSS problem

In 2000, the prognosis was poor (without some new approaches) for solving the issue that when implementing an OSS, a CSP wants it to work with the ones it already has, leading to high systems integration cost and complexity.

In 2021, we find this still a problem, but a significantly reduced one because many OSSs are being encapsulated with good APIs, refactored, or replaced with cloud-native software. Cloud-native software technology, in concert with CI/CD development processes and deployment on private, public, or hybrid clouds, provided just the new approach that was needed to reduce—although not completely solve—the legacy OSS problem.

The plug that won't plug in

In 2000, we were in a sorry state, with OSSs unable to plug into multiple vendors' network elements or into other OSSs. Proprietary command lines still dominated while EMSs nearly always had to come from the network element vendor. We were playing with CORBA/IDL interfaces but finding that the "C" for "common" in CORBA was not common at all. Prognosis was "very near impossible (now what?)."

Today, the development of NETCONF/YANG interfaces with cloud-native software architecture is democratizing the interfaces, making it easier to interface the domain control systems (the new generation of EMS/NMS) to the (often virtualized) network elements. Fully functional multi-vendor interfaces are still rare, but the rise of practically oriented standardization groups such as the Telecom Infra Project (TIP) thatare choosing good, practical specifications and driving them into implementation are making reasonable progress in multivendor element support. Northbound interfaces from the domain controllers to upper-layer orchestration and OSS systems are, similarly, being specified and implemented. Interfaces between OSSs are still not well standardized, but the rise of cloud-native architectures and open API management tools have at least made it possible for element and OSS vendors to provide cheaper reasonably functional and open APIs to integrate among the OSSs.

Too much data, not enough information

At the turn of the century, a network database was a static thing that had between 30 percent (core network) and 70 percent (outside plant) bad or outdated records. Processes to clean the data and reconcile issues between databases were expensive and time-consuming. Few inventory systems were open to others for queries or synchronization with anything more

sophisticated than database dumps and ETL tools. We were starting to synchronize the databases with the network elements through auto-discovery but had the problem that we were using up too much of the precious computing power of the network elements. Prognosis was good for the new generation of OSSs, but poor for legacy systems.

In 2021, all modern network elements, whether virtual or physical, can be automatically queried, or even announce themselves to the northbound systems. They also have the processing power to announce changes in configuration, or even real-time state. Domain controllers are in play that automatically synchronize with these elements and make the information freely available to other systems. We pretty much solved this one.

Falling hardware costs and the software shift

While hardware cost per unit of capability dropped by nearly twelve orders of magnitude between 1975 and 2000, software cost per function point only dropped by less than three orders of magnitude. Each transition to new software technology, such as object-oriented programming and reusable architectures, including J2EE and .NET, generally contributed about a 20 percent reduction in cost to produce and maintain. But there was a fair prognosis to bring this down further with web-based user interfaces, component-based software technologies, and better system-to-system interfaces coming into focus.

The rapidly decreasing cost of the underlying computing hardware combined with the generalization of service-oriented architectures to include many internal APIs (not just APIs to external systems) led to the creation of cloud-native software architectures. In these, software is broken up into small separately defined, developed, and deployed microservices that have defined APIs between them—kind of a super-SOA architecture. Yes, it is incredibly inefficient in computing resources. But it is so much easier to build, test, deploy, and maintain that this approach is well worth it. This is the single most important technology change in the history of software: separate software components that can be specified, developed, deployed, and evolved easily, decreasing the cost by about an order of magnitude. Further advances in service meshes, simplifying the communications infrastructure between the microservices, and integrated security are further reducing the cost and time to develop, deploy, and evolve the microservices.

Today's challenges

With these major advances from the last 20 years, have we solved all the major challenges in OSS today? No, we have new challenges today and into the future. But the prognosis is good.

Managing disaggregated network elements

In the optical, packet, and radio technologies, work is underway to break the larger network elements into smaller pieces that can come from multiple vendors. These hardware and software pieces have specified interfaces and, in many cases, also have defined requirements to allow them to be put on open bid as "white box" solutions. Disaggregation is happening in both the horizontal (breaking boxes up into multiple boxes with the communication path threaded

through them) and vertical (breaking them into the hardware and separate control software). For OSSs, this means a larger, more complex set of boxes to engineer, install (whether virtual or physical), configure, and assure, with multiple vendors involved. Fortunately, the specified interface technology and specifications discussed before can help considerably. And the inability to create out-of-the-box plug-and-play interfaces (an impossible problem to solve) is mitigated by cooperative testing programs among the piece parts in multi-CSP standards groups.

Effectively managing virtualized network elements

Network function virtualization (NFV) is proceeding in CSPs with software-based virtual network elements (running on data center hardware) replacing hardware solutions, but only slowly. In contrast, in enterprise data center routing, virtual routers already constitute over half the market. The cost and management complexity of the computing and storage infrastructure has slowed deployment in CSP networks, but good solutions are now available from companies like HPE, Red Hat, and VMWare. However, the majority of the VNFs are still being managed as virtual boxes as before, with little use of the full potential of the technology. These need to be tied into the new software-controlled networking technologies to reach their potential to create an elastic network, automatically adding new network capacity at the right time at the right place.

Implementing software-controlled networks

Command line and other manual interfaces to network elements (whether virtual or physical) still are state-of-the-art for many CSPs in many technology areas. However, certain areas, such as SD-WAN, are rapidly incorporating software control. In SD-WAN, enterprise customers have portals with information and direct control over most of the resources that support their service. But this needs to be spread among the other network technical areas with sets of more sophisticated domain controllers and cross-domain orchestration systems.

Securing the network, services, and devices

Security of the network and computing and storage resources themselves as well as the services riding on them and the premises equipment attached (IoT devices, in particular) has become a major concern among leading-edge companies as workspaces have radically distributed due to the necessity to work at home. Today, security is implemented more as an add-on, or even afterthought, than an intrinsic part of most network operations. This needs to change.

Supporting complex network slicing

Network slicing, where parts of the shared network are dedicated to specific customers, services, or QoS guarantees, must be done end-to-end in the engineering, provisioning, and management of the network. This requires that all the OSSs be coordinated across all the technological, geographic, and organizational domains. This is not only a technical but also a planning, operational, and budgetary issue for OSS.

Evolving to autonomous networks

Various groups are defining what the autonomous networks of the future will be, when they are virtualized to the greatest extent, put under software control, and controlled via ubiquitous artificial intelligence. Besides the problem of the sheer magnitude of implementing this vision across the entire network, there is a fundamental problem in trusting the actions of the AIs. Individual AIs themselves can determine what to do but are incapable of explaining why they choose that course of action. Explanation AI (XAI) technology that will solve this problem is in its infancy. And even if individual AIs work well in their individual domains, the autonomous network requires multiple interacting AIs in their various domains, with their differing goals. Some ring-fencing of their cooperative, global actions is needed to prevent black swan events. Also, probably sandbox simulations on digital twins of the network will be required to check on the efficacy and advisability of the recommended actions. Much work is needed here.

The path behind and journey ahead

Since the beginning of the century, significant progress has been made in OSS technology and functionality. But much more is needed as we move to autonomous, intent-based networking. And as the communications infrastructure is merged into the computing and storage infrastructure, new management challenges will present themselves. It has been, and is going to be, an interesting ride.