An Architecture for Network Operations and Management Based on State and Services

John C. Hoag

McClure School of Communication Systems Management, Ohio University
297 Lindley, Athens, OH 45701
hoagj@ohiou.edu
http://www.csm.ohiou.edu/sbnm

Abstract. This paper describes an architecture for network management that bridges the domains of operations support systems (OSS) and network monitoring systems (NMS) by means of information grid concepts and web services practices. Central to this architecture are the expanded definition and use of system state and the expanded reliance on semantic reasoning. Emerging requirements for network management come from U.S. defense needs and emerging standards such as GMPLS. The recent availability of open and collaborative tools will accelerate the development of the system described herein, known as State-Based Network Management.

1 Introduction

In an era of convergence and an environment of competition, the need for efficient network operations has never been greater. The gap between expectations for systems and the capabilities of components is significant but could be mitigated through a fresh look at requirements and adoption of new techniques. The ultimate expectation is for a provider or enterprise customer to allocate and reallocate resources as a function of dynamic requirements, mobility, and availability of network elements. The enabling technologies for timely development include web services, open systems, and semantic reasoning.

2 Architecture Overview

The notion of state in networking has been seen as overly simplistic and an impediment to scalability. Datagram routing, for instance, is a memoryless function that treats successive packets independently even if they have a common destination. We have held strongly to the position that networks not be application-aware and vice versa. From another perspective, network element and configuration data, which also constitute state, have been rigidly maintained in potentially non-accessible, non-interoperable, information *silos*. This research posits a more inclusive notion of network state: the current value of each of the following models:

- The set of all application requirements per endpoint, aggregated only for similar quality of service
- The set of all network elements in provider(s) inventory and their general availability
- The set of all provisioned (i.e, assigned) resources, which is a cross-reference of the previous two items
- The set of generic templates for configuring network monitoring all classes of devices and elements

In the proposed distributed architecture, agents at local nodes such as enterprise customer endpoints and provider edges must maintain their own local state.

The architecture proposed by this research introduces a centralized state engine whose objective is to exactly fulfill the current set of telecommunications requirements by acquiring provider resources and monitoring them. This can be considered a *bridge* function between the two domains of network management: operations and business support systems (OSS/BSS) and network monitoring systems (NMS). Moreover, this engine is proposed to automate previously manual tasks based on an emerging semantic web approach being developed along with defense and homeland security applications collaborating with this project.

The notion of an information grid not only implies meshed interconnection, it also infers interoperability among collaborative systems. The current network management paradigm, polled and driven by events, has been organized around centralized intelligence, perhaps organized in a hierarchy of element managers. The increase of processing power and memory in network devices has not been used to alter this paradigm. We propose to inject concepts from the information grid to the management plane based upon the following [1]:

- Insightful messaging between nodes
- Local state maintained within each node
- Local resources, functionality, and self-awareness
- Ability to coordinate passage of transit traffic

The syntax and semantics of inter-node communication are arguably the biggest challenges in development of an information grid in any domain. The key enabler for this project is a service-oriented architecture and a semantic web created for the telecommunications management domain, which implies common metadata and a common ontology accepted by participants [2]. In either the management or control planes, the ability for autonomic devices to carry out their full spectrum of duties without operator intervention is the long term vision of this project.

Whereas in the current telecommunications paradigm, information is pulled into the network manager for action, this research embraces "smart push" between the NMS and the state engine. One motivation for this is the mitigation of the problem of massive (and duplicated) data sent to managers. The volume of messages increases all forms of latency: congestion delays, queuing delays, processing delays in the host operating system, and processing delays in the management application itself. Event correlation and filtering, in this design, remain delegated to the element manager software, but responsibility for topology, state, and action will reside in the state engine model.

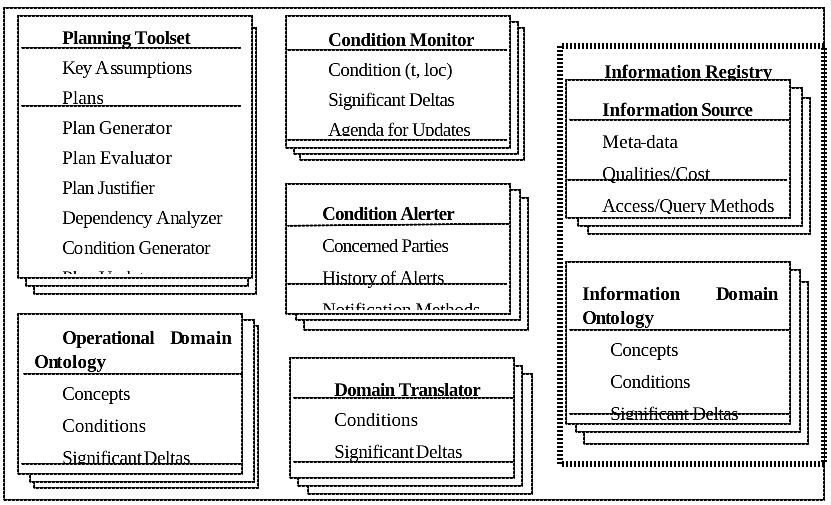

Fig. 1. Graphical representation of VIRT. The initial use case for VIRT was delivery of aviation weather enroute, hence the special and temporal reference

The formal model for this "smart push" implementation is called Valued Information at the Right Time (VIRT), whose original motivation was lessen the task of high-consequence operators such as pilots, who often receive burdensome volumes of raw data that are often late and cannot be used to adapt the mission in progress [3]. VIRT is project led by Hayes-Roth within the Worldwide Consortium for the Grid (W2COG), sponsored by U.S. defense executives.

VIRT incorporates a broad notion of state that contains information about both goals and conditions, with the capability to obtain sensory information, to provide ontology-based translation on its inputs, to derive the difference between plans and conditions, and to recommend changes to the current plan.

While most details in Figure 1 are somewhat self-explanatory, the planning toolset merits special attention. The VIRT process can be viewed as always alert and able to execute a procedure when a state-changing event is detected. The basis for logical reasoning within VIRT is both the current plan and the justification for selection of that plan with respect to assumptions. Within VIRT, a set of semantic relationships rules would be developed that concern the stability of the assumptions and the dependency of the plan upon them. In the author's use case, assume that a flight mission (i.e., plan) requires the use of certain navigation instruments, hence the selection of a certain airplane. If the instruments or the assigned aircraft are unavailable, the mission planner must be alerted – and perhaps VIRT can recommend an alternative plane, another destination, a delay for better (e.g., VFR) weather, etc.

Defense and homeland security expectations are driving new requirements for network operations and management. The U.S. Defense Information Systems Agency (DISA) has engaged a pilot project to demonstrate a "service-based, commercially-managed, and commercially hosted" model known as Net-Centric Enterprise Services (NCES) [4]. These ad hoc and persistent applications and infrastructure capabilities are considered most valuable by the Government:

- Session Management
- Text Messaging within a web conference
- Whiteboarding
- Application Sharing
- Audio (within a conference session)
- Presence and awareness (of others attending)
- File Transfer
- Voting and Polling
- Video, shared live or playback
- Recording
- Instant Messaging
- Chat rooms
- Virtual Spaces (for asnynchronous collaboration)

Moreover, this model is intended to apply to U.S. and coalition forces, for classified and classified purposes, and for the spectrum of intermittent through well-connected users.

Equivalent cases can be constructed for network management. We will consider two cases involving, respectively, new and existing services, with respect to the reference architecture presented in Figure 2. The first case is to turn up a new WAN service (e.g., ISDN PRI) between a pair of customer endpoints, where a U.S. competitive local exchange carrier (CLEC) is obtaining facilities from its incumbent carrier. The second case a circuit outage for some future enterprise customer that can manipulate virtual resources to restore services.

The drawing below represents the information flows between components within the architecture proposed by our research. Nodes will be discussed below, and information flows are labeled by their protocol (e.g., XML or SQL) and an identifier that will be used to describe the use cases.

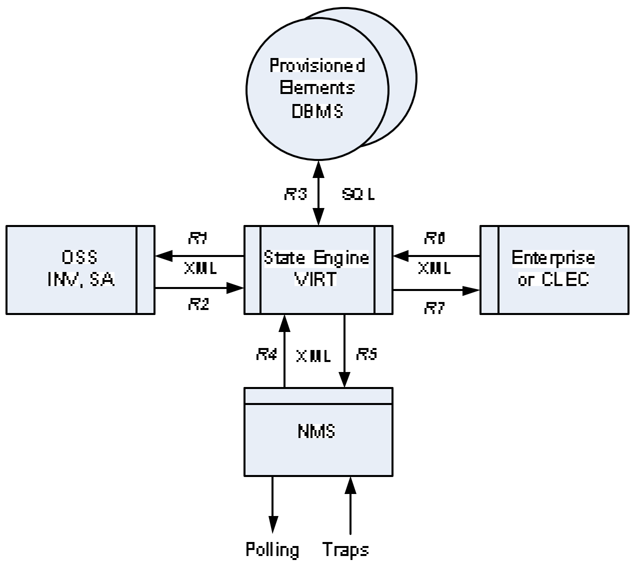

Fig. 2. Graphical representation of State-Based Network Management

In Figure 2,

- The central engine maintains state as defined previously and executes the VIRT process – basically to monitor conditions in order to maintain state, and to obtain resources necessary to fulfill requirements.
- The rightmost box represents the agent acting on behalf of the customer. In the management grid, it manages information about users, applications, and locations and must transact with the central engine in terms of aggregated demand and service quality.
- The upper node represents the database local to the central engine. It is currently depicted as a SQL relational database, but we anticipate the availability of native SOA-complaint software.
- The leftmost box represents the collective OSS/BSS capability of one or more providers. The scope of these interfaces is intentionally limited in our subsequent examples.

The bottom node contains the NMS for monitoring and event correlation. This icon represents use of SNMP as well as perhaps a subtree of other element managers. Northbound messaging via XML is expected to be filtered and terse, passing information only about faults with high confidence

In the following two use cases, the order of the steps is significant. Each step will refer to some link identified by a reference number, which will not be used sequentially.

Consider the case of a non-facilities based CLEC that has sold an ISDN PRI circuit to some customer in order to deliver 23 discrete bearer channels for telephone communications. One endpoint is the customer premises, and the other is a point-of-presence. In this case, the CLEC is the is the network operator and the operator of a VIRT engine

1. CLEC captures information about order, location, and application, then characterizes its use in QoS terms such as delay and jitter
2. CLEC sends this request to its VIRT engine on R6
3. VIRT enters this in the DBMS over R3
4. VIRT prepares an service request message and sends it to its incumbent carrier that actually owns the facilities – local loop, etc. via R1
5. To simplify this example, in this step the LEC performs inventory, preorder, order entry, provisioning, service activation and testing – then promptly replies to the CLEC on R2. We would more correctly divide these into separate steps.
6. Upon receipt, the VIRT engine shall
    a. Update its local DBMS via R3
    b. Prepare information on new managed objects and send to the NMS on R5
7. Functionally, our task is complete. Optionally, an acknowledgment could be sent over R7 to another in-house system to commence billing, etc.

Consider some enterprise that subscribes to some virtual circuit (e.g., Frame Relay, ATM, MPLS) in addition to transport services. Let one or more VCs have an outage.

1. Either polling will reveal the outage or some interface will generate traps the NMS. Events correlate such that the outage is scoped and confirmed. The NMS alerts VIRT over R4.
2. The VIRT engine queries its DBMS on R3 to determine location, QoS, and application information.
3. At this point, VIRT can invoke steps 4-5 above.

3 Evaluation of Concurrent Initiatives

Many concurrent initiatives are developing components that have been identified for use in the above-proposed architecture. This section will summarize the design, contribution, and status of each of the following: OpenOSS, OSS/J (Java) , OpenNMS, and KING. The first two projects listed incorporate the efforts of the Telemanagement Forum, a consortium of vendors; the two "open" projects make source and executable files available through public licenses. Open source development implies uninhibited free access to source code and collaboration among self-selected developers; technical support ranges from ad hoc and voluntary to, such as Red Hat Linux, professional and paid.

The OpenOSS Initiative is an instantiation of the Telemanagement Forum's reference architecture for Next Generation Operations Support Systems (NGOSS), specifically in regard to its service assurance functionality. In order to serve as a "catalyst" for acceptance of NGOSS, collaborators have recently demonstrated a proof-of-concept use case that incorporated the fault management (FM) and trouble ticketing (TT) functions using a service-oriented architecture [5]. The SOA interfaces and schema had been developed by another project described herein, OSS/J. The case study was to discern Voice over IP (VoIP) faults, exceeded thresholds, and performance issues and consequently open a trouble ticket. This demonstration involved academia, vendors, and a major carrier. It successfully integrated a variety of commercial off-the-shelf (COTS) and open source software packages including OpenNMS, which will be described shortly [6].

The OSS Through Java Initiative (OSS/J) is a consortium of OSS vendors and infrastructure firms that has developed a standard implementation of the Telemanagement Forum's Next Generation OSS architecture [7]. This consortium has produced detailed interface standards and is tasked with certifying products' compliance with them. Interfaces are specified using WSDL, the web services design language, and are to be transacted using a consortium-designed XML schema. OSS/J enables interoperability by sharing details of its interfaces but does not incorporate the open source model of licensing because its members produce COTS. Although interfaces define message structure and semantics and are platform-neutral, their development and certification processes involve Java J2EE. Four interfaces have reached the stage of final release: Service Activation, Quality of Service, Trouble Ticketing, and Inventory. The draft Fault Management and Trouble Ticketing interfaces were successfully implemented in the OpenOSS catalyst demonstration.

OpenNMS has a much different scope of functionality and has also been successfully implemented in the OpenOSS *catalyst* demonstration [8]. Its functions specifically include service polling, data collection, and event management. Of specific interest to this research is he fact that OpenNMS definitions for managed elements are expressed in XML format; this implies that an outside agent could add, modify, or delete configuration files. OpenOSS is a classic open source endeavor, written largely in Java, built for a variety of Unix platforms, and with over five thousand downloads of the most current stable release.

System documentation indicates that OpenNMS can poll over twenty thousand devices and over two hundred thousand datapoints every five minutes, but this is constrained by the process of posting each event to an SQL database [9].

The KING Next Generation Network Architecture is an exemplar of adaptive, automatic, and autonomic network resource allocation based on centralized intelligence [10]. This design seeks optimal performance including quality of service and has the capability to regulate admission controls and policy-based routing. This design resembles management of the U.S. electricity transmission grid in that a recurring batch process discovers network topology, obtains flow and blocking information per link, and subsequently adjusts admission control and router forwarding policies – balancing the demand for network resources with capacity [11].

Their simulation results suggest that traffic blocking can be halved through KING techniques. Stated differently, KING results imply that network performance varies widely relative to admission controls and forwarding policies in a QoS-aware setting. A consequence of this observation is that optimal network management may depend on additional algorithms and state variables that are not currently incorporated in either OSS or NMS suites.

The preceding four analyses refer to present infrastructure, services and protocols; in this setting, effective routing and transmission protocols can mask both the issues with underlying transport and the corresponding management effort. In an irreversible trend, these approaches will be displaced by flexible packet and frame -based transport and virtual services [12]. Generalized Multiprotocol Label Switching (GMPLS) is a superset of MPLS that offers an open, standards-based feature -rich control facility for virtual private networks (VPNs). U.S. carriers have migrated to IP backbones and have extended the edge of their MPLS networks to their customers, who potentially have great advantage in managing their own tag spaces within spanning trees established by their providers. While standards efforts are not yet complete, expectations are rising for enterprise customer control over Layer 1 as well as lower Layer 1 recurring charges.

The significance of packet-over-SONET and GMPLS is, much like the DISA NCES model mentioned previously, that expectations for OSS/BSS and NMS integration are central to the deployment of advanced services.

4 Conclusions

With respect to the previous four systems, the VIRT-based, state-based network management (SBNM) model advanced by this research has a much larger scope. SBNM serves to bridge the OSS and NMS domains by means of a common ontology and semantics as well a service oriented XML interfaces wherever practical; even the interface between the realtime NMS and the VIRT engine in each direction can benefit. The presence of various open initiatives does not diminish the need for this state-based bridge, since these tools serve to accelerate development and increase collaboration.

Requirements are driven both by the needs of users such as the U.S. DISA NCES program and the economics of deploying efficient services such as GMPLS. These are both strong motivators for network managers to look beyond integration of known services and to engage with efforts on emerging applications and infrastructure.

Since the objectives the systems under comparison differ, further analysis will reveal where their architectures differ as well. The Software Engineering Institute [12] indicates that, to compare candidate architectures, simulation and experimentation are preferred over rigid use case scenarios and fine-grained metric development. Specifically, our future work must seek to find the balance between management based on web service transactions (which will be detailed but rare) and that based on events (which will be terse but voluminous). Characterization of these models in detail will permit simulation and experimentation, permitting us to estimate the delays involved in the network management.

The modest size of the telecommunications network management market is not large enough to drive new software paradigms. In concert with the rapid bottom-up open developments, a fresh top-down look at both requirements and architecture will be good. Incorporation of concepts from the information grid coupled with the semantic web will help organize our future work.

This paper has contributed to the understanding of many concurrent initiatives that seek to improve network management. This paper uniquely has attempted to justify the use of state modeling and dispel concerns about unintended consequences. Most importantly, this paper has offered both an expanded set of requirements – application specific, QoS-aware, on-demand – and an architecture to fulfill them.

References

1. Foster, Ian, "What is the Grid? A Three Point Checklist," Grid Today, October 2002, http:/www-fp.mcs.anl.gov/~foster/Articles, accessed January 12, 2006.
2. Hayes-Roth, Richard, "Model-Based Comunication Networks: Filtering Information by Value to Improve Collaborative Decision Making," Naval Postgraduate School, Unpublished Manuscript, June 23, 2004.
3. Hayes-Roth, Richard, "Model-Based Communication Networks and VIRT," Presented to the 10th International Command and Control Research and Technology Symposium, 2005.
4. Defense Information Systems Agency, "Net-Centric Enterprise Services Collaboration Service Request for Quotation, Statement of Objectives," Attachment 2, December 16, 2005..
5. Gallen, Craig, and Jeffrey Reeve, "Investigating the Feasibility of Open Development of Operations Support Systems," in Proceedings of the 9th IFIP/IEEE International Conference on Integrated Network Management, 2005.
6. British Telecommunications plc, "Open OSS: A New and Complementary Approach to Drive the Implementation of OSS Standards," http://www.tmforum.com Document Library, White Papers, accessed January 12, 2006.
7. "OSS Through Java Initiative, The Only Game in Town," http://www.ossj.org/learning/whitepapers.shtml, accessed Jan. 12, 2006.
8. "OpenNMS," http://www. opennms.org/wiki/, accessed Jan. 12, 2006.
9. Balog, Tarus, "Enterprise-Wide Network Management with OpenNMS," O'ReillySysAdmin, http://www.oreillynet.com/pub/a/sysadmin/2005/09/08/opennms.html, accessed January 12, 2006.
10. Walter, Uwe, and Joachim Charzinski, "Architecture of a Network Control Server for autonomous and efficient operation of Next Generation Networks," presented at the 13th International Conference on Telecommunications Systems Modeling, 2005.
11. Hoag, John, " From the Electricity Grid to the Telecommunications Grid: A New Paradigm for the Network Management Application, " presented at the W2COG Working Symposium, 2005.
12. Takeda, T., et al, "Layer 1 Virtual Private Networks: Driving Forces and Realization by GMPLS," IEEE Communications Magazine, July 2005.
13. Aboud, Gregory, et al, "Recommended Best Industrial Practice for Software Architecture Evaluation," Software Engineering Institute, TR96-025, 1996.