

Network Simulation for DODAF-Compliant Architectures

J.A. “Drew” Hamilton, Jr., Ph.D.
Computer Science & Software Engineering
Auburn University
(334) 844-6360
hamilton@auburn.edu

This paper will describe how DOD Architecture Framework (DODAF) – compliant architectures can become the basis for operational network planning tools for the Combatant Commands through the use of executable architectures.

Introduction

An “executable architecture” is defined as the use of dynamic simulation software to evaluate architecture models. The system attributes from a DODAF-compliant architecture can be used to directly load a network simulation tool thus producing an executable architecture. Such an executable architecture can be used to validate the operational and system views and check the internal self-consistency of the DODAF-compliant architecture.

The mandatory use of the DOD Architecture Framework is prescribed in DOD Instruction 5000.2, in which proponentcy for operational views is assigned to the Joint Staff, while the Under Secretary of Defense (Acquisition, Technology, and Logistics) (USD(AT&L)), leads the development of the system views in collaboration with the Services, Agencies and Combatant Commanders [1].

As defined in the DOD Architecture Framework [2], an Operational View (OV) “is a description of the tasks and activities, operational elements, and information exchange required to accomplish DOD missions.” A Systems View (SV) is a set of graphical and textual products that describes systems and interconnections providing for, or supporting, DOD functions. The SV associates systems resources to the OV.” The DODAF is a critical part of the Joint Capabilities Integration and Development System (JCIDS) as defined in CJCSI 3170 [3].

Section 7.3 of volume I of the DODAF outlines the concepts behind an executable architecture and specifically states that the architecture data elements and the attributes required to construct executable models are specified in volume II of the DODAF. Our approach is slightly different; we are focusing on automated tools to build as-is architectures and then directly loading a network simulation topology from the architecture. Hence we are working from a system-centric view. By “as-is” architecture, we mean the architecture of existing systems as opposed to future or “to-be” architectures. Because our focus is on existing systems and networks, this application of the DODAF warrants our systems view orientation.

Network Simulation Enables Executable Architecture

By modeling an existing network in the form of an “as-is” architecture, we can create a simulation model, which when stimulated with appropriate traffic, can be an executable architecture. Intuitively, the best way to plan for the operation and performance of a network is to observe and measure the results. This is impractical in a dynamic environment when assessing alternate strategies for communications asset employment. Network simulation offers a realistic alternative for J6/G6 planners.

The three major challenges in network simulation are: network monitoring, modeling the network and operating and maintaining the simulation [4]. We are developing/modifying automated network discovery tools to build the DODAF architecture views necessary to load a simulator. Open source network topology builders can be modified to generate Systems Interface Descriptions (SV-1), and Systems Communications Descriptions (SV-2). Monitored network traffic can be used to generate the Systems-Systems Matrix (SV-3) and the Systems Data Exchange Matrix (SV-6). The SV-3 provides the who is talking to who information and the SV-6 provides the message characteristics such as length and format. Finally, we need to know the physical characteristics of the physical communications devices. Hopefully, this information will become readily available as new communications device acquisitions use the DODAF to document their systems. Though the number of DOD communications devices in use is large, that number is finite and can be manually modeled as a Systems Performance Matrix (SV-7). These three sources for building the model are outlined in Figure 1.

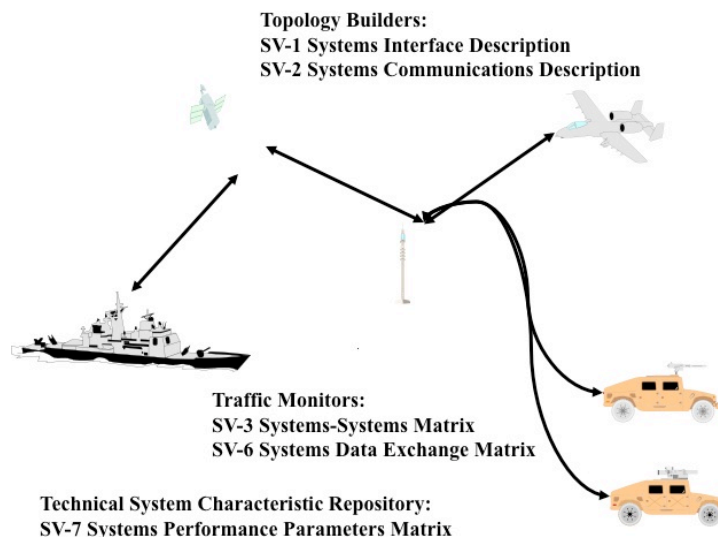


Figure 1. Building Automated System Architectures.

Once the minimal set of five systems views are developed and loaded into a simulator, operational planners can vary locations, workloads and configurations and perform “what if” analysis on their communications plan [5]. A network simulation derived from a DODAF-compliant architecture has multiple uses. Operational concepts can be

evaluated via the simulation. More importantly, such an executable architecture can address throughput and utilization issues not directly covered in static DODAF system and operational views.

Not only can DODAF-compliant executable architectures support operational planning; executable architectures provide the critical third leg needed to validate a DODAF architecture.

Validating a DODAF-compliant Architecture Through Network Simulation

Since the current Defense Acquisition System [1] mandates the use of the DODAF, developers have strong motivation to demonstrate that their architectures are valid and internally consistent. A model is valid if it can produce the outputs that are equivalent to the ones that would be observed given the same inputs in the environment being modeled. In other words, the model is capable of predicting the behavior of the system being modeled within a specified tolerance. The precise meaning of equivalent inputs and outputs will vary depending upon the simulation.

The term verification describes the activity of insuring that a particular implementation faithfully satisfies the requirements of a specification over a given range of inputs. If it is known a priori that a model is valid, then historical data may be used with the understanding that demonstrating a correspondence between program outputs and the physical phenomena being modeled implies the verification of the simulation program in question.

In addition to systems views, the DODAF defines Operational Views, Technical Views and All Views. Operational Views are fully defined in Volume 2 of the DODAF [6]. Succinctly, Operational Views (OVs) are representations of requirements. Consequently, there is a direct relationship between OVs and SVs. Figure 2 is a variation of Knepell and Arangno's validation structure [7] adapted for application to the DODAF.

An operational concept is not valid if it cannot be supported by the systems available in theater. So in this sense, the systems views validate the conceptual model of the operational view as shown in Figure 2. Conversely, the validity of systems architecture can be evaluated against how well it supports the requirements documented in the operational architecture.

The employment of an executable architecture adds a new and needed dimension to the verification and validation of a DODAF-compliant architecture. Executable architectures can assess the validity of an operational concept. While the SVs may provide the needed

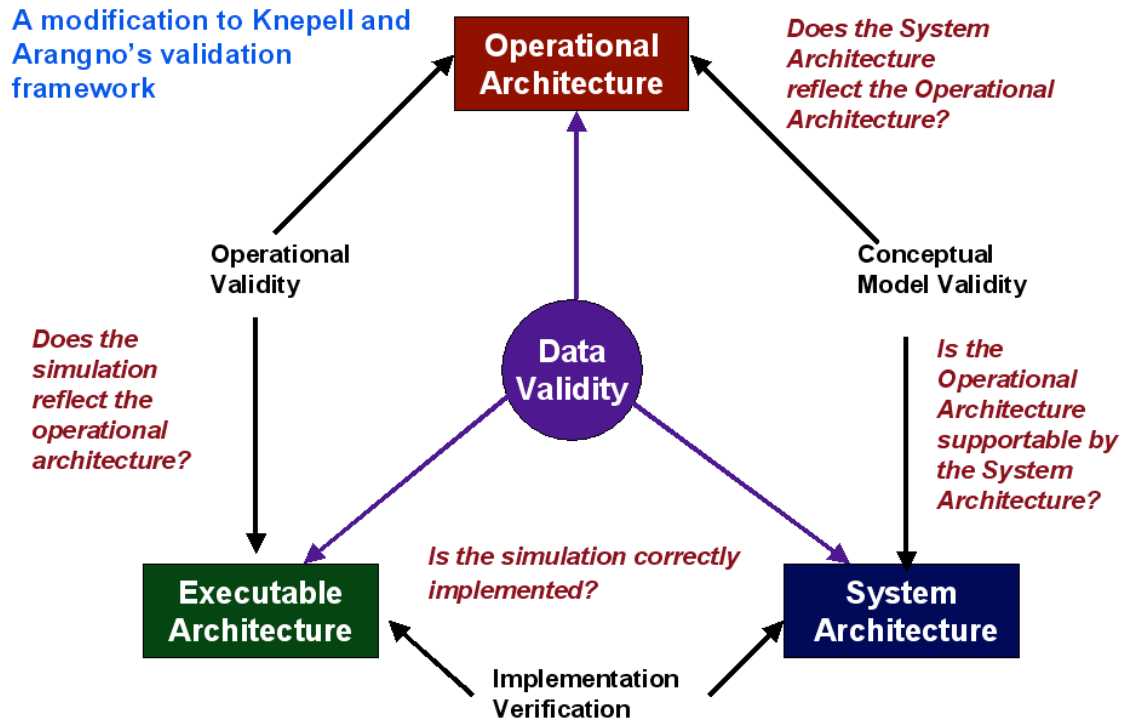


Figure 2. Validating a DODAF Architecture.

connectivity to support the operational concept described in the OVs, the SVs alone do not give sufficient insight into meeting operational performance and capacity needs. It can be argued that required performance can be extrapolated from the SVs, but an executable architecture can provide a much more dynamic and flexible means of evaluation.

Executable architectures can validate the operational concepts in the OVs by exercising the system architecture with realistic traffic loads. It is straightforward to create traffic scripts based on traffic data collected from similar exercises conducted on the many instrumented ranges in the DOD. This allows for the

Finally, an executable architecture can verify the implementation of the system views. It is common in the network simulation domain to use the simulation to debug its underlying model and vice-versa.

Operational Use of Executable Architectures

One of the chief obstacles to the operational use of network simulation has been the long lead times needed to set up and run a validated network simulation. Automated tools such as Nmap (<http://www.insecure.org/nmap/>) and Netdisco (<http://www.netdisco.org/>) can be used to load a network simulation. Since these and other tools are open source, they can be modified to build out a system architecture that is both DODAF-compliant and capable of loading a network simulator.

The DODAF itself is also a key enabler for the operational use of network simulation. As more and more acquisition programs comply with the requirements to document their systems in accordance with the DODAF, it is reasonable to assume that there will be system performance data formatted as SV-7s readily available. The NETWARS Project (http://www.opnet.com/products/library/netwars_models.html) demonstrates the feasibility of collecting military and relevant commercial system models formatted for quick inclusion into a network simulation.

Heretofore, we have not addressed the data aspects of the DODAF. Data validity is the center of Figure 2 because common, valid data elements are core to the DODAF. However, this is too great a level of detail for operational planning. Also, we have not focused on the operational views for several reasons. Combatant Command staffs fully understand the doctrinal templates that underpin operational views. Different missions will have different operational views for the same units. Our approach focusing on systems views to build executable architectures is applicable to any mission area.

One practical example of using executable architectures to support operational planning involves defending against distributed denial of service (DDoS) attacks. A denial of service attack floods a network with so much traffic that legitimate traffic is blocked. This is analogous to jamming a radio network. A distributed DoS attack is one that is launched from many stations instead of a single station. Mirkovic and Reiher classify DDoS defense mechanisms as preventive, reactive or autonomous. An executable architecture can be used to evaluate each type of mechanism. One prevention strategy is to place “forward deploy” firewalls on the outbound ports of the main routers as described in [9]. The performance impacts of various firewall configurations and placements are readily displayed through an executable architecture. A typical reactive strategy is to simply reconfigure the network and reroute traffic to a server that is (hopefully) not under a DDoS attack. One autonomous means to mitigate a DDoS attack is to use a dual-queue system, which automatically starts dropping traffic that comes from untrusted hosts at the onset of an attack [10]. All of these partial solution strategies to defend against DDoS attacks can be systematically evaluated through an executable architecture.

Conclusions

At Auburn University, we are working with monitoring and simulation tools developed in the research community to develop automated architecture builders with direct feeds into network simulators. In many cases source code is readily available which provides the capability for extensibility and better understanding. Also, our research group’s work in software vulnerability analysis leads to the conclusion that it is easier to evaluate source code for security flaws than to evaluate only the compiled code [11].

In applying the DODAF to operational communications planning, our research group takes a “systems view first” approach. It should be noted that many architects take an “operational view first” approach. The utility of either approach (and other alternatives)

depends upon how the DODAF is being applied. Referring again to Figure 2, we propose building the executable architecture from the system views and then using the simulator to evaluate the ability of a communications plan to support a specific operation. This is consistent with many approaches to operational view development that are often based on specific mission threads.

The DOD Architecture Framework is another example where the DOD is out in front of when it comes to large-scale information architecture. Mainstreaming the DODAF into the Combatant Commands will provide the Combatant Commands with a standard, useful tool to support operational communications planning.

References

1. DOD Instruction 5000.2, Operation of the Defense Acquisition System, 12 May 2003: p 2.
2. DOD Architecture Framework version 1.0 Volume I: Definitions and Guidelines, <http://www.defenselink.mil/nii/doc/>, 9 Feb 2004: p 1-2.
3. CJCS Instruction 3170.01D, Joint Capabilities Integration and Development System, 12 Mar 2004.
4. Hamilton, J.A., Jr., Nash, D.A. and Pooch, U.W., Distributed Simulation, CRC Press, Boca Raton, Fla., 1997: p 331.
5. Hamilton, J.A., Jr., Modeling Command & Control Interoperability, SCS Press (<http://www.scs.org/>), San Diego, Calif., 2004: pp 85 – 99.
6. DOD Architecture Framework version 1.0 Volume II: Product Descriptions, <http://www.defenselink.mil/nii/doc/>, 9 Feb 2004.
7. Knepell, P.L., Arangno, D.C., Simulation Validation, IEEE Computer Society Press, Los Alamitos, Calif., 1993.
8. Mirkovic, J., Reiher, P., “A Taxonomy of DDoS Attack and DDoS Defense Measures,” ACM SIGCOMM Computer Communications Review, vol. 34, no.2, Apr 2004: pp 39 – 54.
9. Chatam, J.W., Using Strategic Firewall Placement to Mitigate the Effects of Distributed Denial of Service Attacks, Thesis, Auburn University, Aug 2004.
10. Fletcher, H.W., Eoff, B., “Braving the Storm: Maintaining Connectivity in the Face of a DDoS Attack Using Trusted Hosts.” unpublished manuscript.

11. Hamilton, J.A., Jr., Greaney, K.J., Evans, G., "Defining a Process for Simulation Software Vulnerability Assessments," CrossTalk, vol. 16, no. 11, Nov 2003: pp 22-25.

Biography

John A. "Drew" Hamilton, Jr., Ph.D., is an associate professor of computer science and software engineering at Auburn University and director of Auburn's Information Assurance Center. He is the President of the Society for Modeling & Simulation, International (SCS), and Vice-Chair of ACM's Special Interest Group on Simulation (SIGSIM), on the Board of Directors of the Alabama Modeling & Simulation Council (AMSC), and Director of Auburn University's branch of the McLeod Institute of Simulation Science (MISS). Dr. Hamilton is a member of the Simulation Interoperability Standards Organization (SISO) and has previously served on the SISO Conference Committee.