Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds.

ARCHITECTURE FOR COMPARING ALTERNATIVE DESIGNS OF A TACTICAL NAVAL COMMAND AND CONTROL SYSTEM USING DISCRETE-EVENT SIMULATION

Shaun P. Hayes

Department of the Navy
Naval Sea Systems Command
Washington Navy Yard, DC 20376, U.S.A.

Eugene P. Paulo

Department of Systems Engineering
Naval Postgraduate School
Monterey, CA 93943, U.S.A

ABSTRACT

Research was conducted recently at Naval Postgraduate School focusing on the development of a system architecture for a tactical naval Command and Control (C2) system. The system architecture methodology started with describing an operational concept, moved to the co-development of the functional architecture and the physical architecture, and concluded with the development of a notional operational architecture. A portion of the contact prosecution process for a Surface Action Group (SAG) involved in securing local sea control was modeled in the discrete-event simulation software Arena®, and this effort is the focus of this paper. Simulations for both potential alternative designs were conducted. Finally, a comparison of the simulation results demonstrated the feasibility of the developed architecture framework to compare alternative designs of the C2 system.

1 INTRODUCTION

As a result of the numerous definitions and concepts of C2 throughout history, the design of a C2 system is a daunting task. Adding to the challenge is the understanding and integration of new operational concepts identified by stakeholders as necessary to meet their operational needs. Network-Centric Warfare (NCW) is one such operational concept which has implications on the design and development of C2 systems.

There has been much work conducted to date to enable NCW by networking combat and information systems and by removing the "stove-pipes" of legacy systems. A military C2 system, however, is more than the technology and equipment which comprise it. It also includes the doctrine, organization, training, leadership, personnel, and facilities surrounding the material. It is imperative for the systems engineer to understand the implications of each portion of DOTMLPF on the lifecycle of the system.

Our research focused on dissecting the complex engineering process of tactical naval C2 systems in order to identify the points of integration between doctrine and material. The goal was to better understand the influence of doctrine on the overall architecture of the material system in order to ensure developing net-centric systems and net-centric doctrine meet the command and control needs of tactical naval forces. Specifically, the focus of the research was on those actions that a future (i.e., ten to fifteen years from present) Surface Action Group (SAG) would conduct to secure local sea control in traditional operating environments. This paper describes the final step of the research, which was to demonstrate the feasibility of the developed architecture framework to compare alternative designs through the use of discrete-event simulation.

2 SYSTEMS ARCHITECTURE

The first phase in the architectural process was development of the operational concept which serves as a general vision of the system from the view of stakeholders (Buede 2000). A key product of the operational architecture was the systems objective hierarchy. The systems objectives hierarchy organized the system's objectives and included the identification of Measures of Merit (MoM) by which different potential designs of the C2 system could be compared.

The second phase in the systems architecture process was the co-development of functional and physical architectures. The purpose of the functional architecture was to describe what the system was to do with the identified inputs to produce the desired outputs. Six top-level functions which a C2 system performs were identified: *Transport Information*, *Process Information*, *Store Information*, *Present Information*, *Generate Response Options*, and *Select Response Options*. These six functions were further decomposed into a hierarchy of sub-functions. For example, *Select Response Options* is composed of three sub-functions: *Evaluate Response Options*, *Judge Response Options*, and *Assign Tasks to Resources*, as shown in Figure 1. The functional architecture also detailed the relationships between the inputs and the outputs of the system.

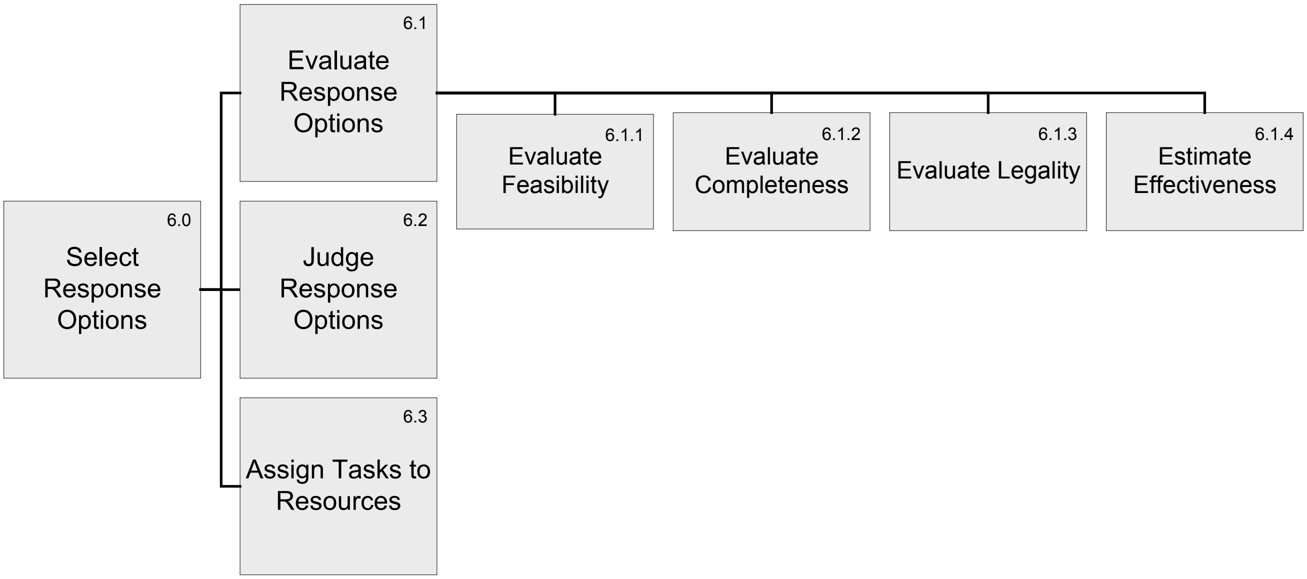

Figure 1: Select Response Option Decomposition

The purpose of the physical architecture was to describe the resources that comprised the system, with resources for every function identified in the functional architecture (Buede 2000). In addition, the physical architecture described procedures by which the system was used (Buede 2000) and controls on the system. Generic components, procedures, and controls were identified. Then, instantiated physical architectures were developed from which two potential alternative designs of the system were created. Before describing the differences between the alternative designs, the impact of NCW on the physical architecture should be discussed.

The concept of NCW consists of a collection of ideas such as geographically dispersed forces, shared awareness, speed of command, self-synchronization, and virtual collaboration (Alberts, Garstka, & Stein 1999). Each of these ideas are impacted by the type of network which facilitates the concept. For example, shared awareness can be accomplished by any centralized, decentralized, or distributed network. Shared awareness, of course, cannot be accomplished through an isolated network. The research conducted followed Baran's (1964) descriptions of centralized, decentralized, and distributed networks. A fourth network description, isolated, was added. All four descriptions are presented visually in Figure 2.

Of the above ideas within NCW, self-synchronization had the greatest impact on the physical architecture. Self-synchronization, as discussed by Cebrowski and Garstka (1998), is the ability of the networked force to "organize and synchronize complex warfare activities from the bottom up." The authority to make decisions and the authority to allocate resources need not follow a traditional top down hierarchy as in a centralized or decentralized network. Decision authority and allocation authority, rather, may be vested in multiple lines of superiors as in a distributed network. In fact, as Alberts and Hayes contend, unity of command is not required for "successfully accomplishing the functions of Command and Control" (2006).

To demonstrate the impact of NCW ideas such as self-synchronization, two alternative system designs of the C2 system were developed. The Alternative #1 design was one with decentralized decision authority and decentralized allocation authority. In contrast, the Alternative #2 design was one with distributed decision authority and distributed allocation authority.

The functions identified in the functional architecture were then allocated to different resources indentified in the physical architecture to form the operational architecture, the final phase of the systems architecture process. Following the function allocation, the functional activation and control structures were defined. As was determined during the operational concept development, one task a SAG securing local sea control must be capable of executing is contact prosecution. Modeling the functional flow, activation, and control of contact prosecution was necessary in the development of the operational architecture. The sub-task of contact prosecution was modeled to serve as an example for future development of the operational architecture.

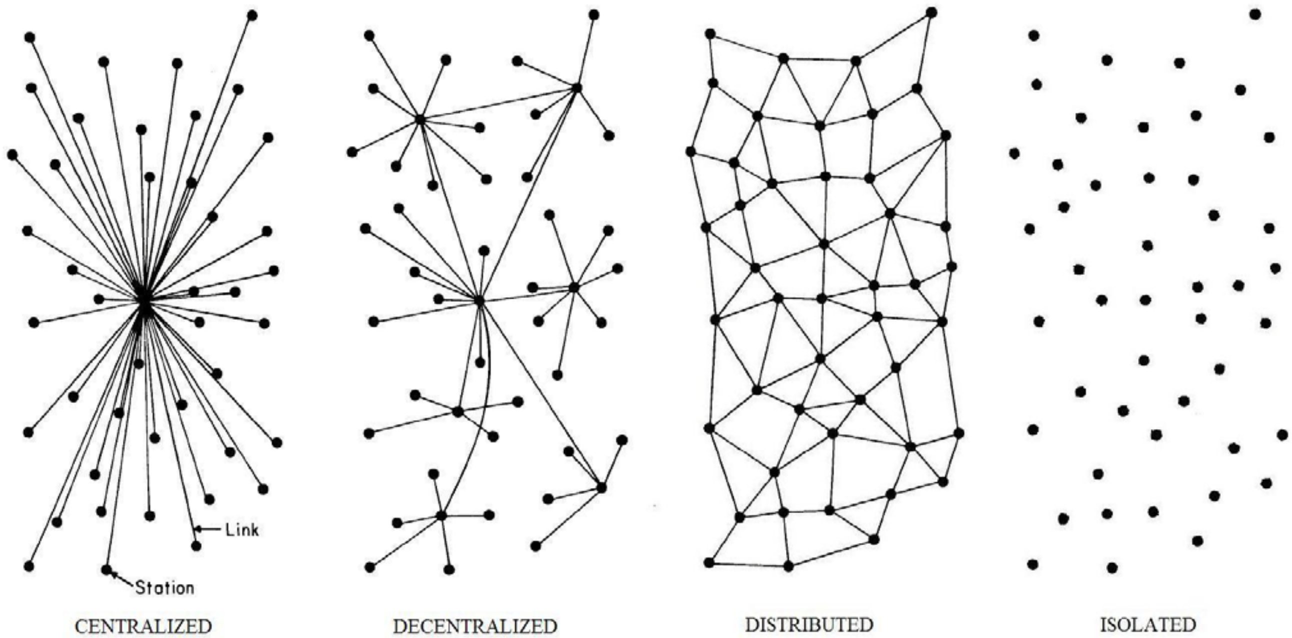

Figure 2: Centralized, Decentralized, Distributed, and Isolated
(after Baran 1964)

3 THE MODEL

The final step in this research was to demonstrate the feasibility of the developed architecture framework to compare alternative designs through the use of the discrete-event simulation software Arena®, version 10.0. Two models were developed, one for each potential alternative design of the system.

In both models, the first step is the creation of a response option. In the Alternative #1, or Decentralized option, three copies of the response option are created and transported using three different links. The first response option to be converted and processed for use is then used as input for the first time through *Select Response Options*. Once a response is selected, three copies are made and are then transported using three different links. The first response to be converted and processed for use is then used as an input for the second time through *Select Response Options*. Finally, an executable response is generated, which marks the end of the Alternative #2 Arena® model. The portion of the Alternative #1 top-level functional flow which was modeled in Arena® is shown in Figure 3.

In Alternative #2, or Distributed option, nine copies of the response option are created and transported using nine different links; three links associated with each of three lines of mechanisms (e.g., commanders) to *Select Response Option*. The first response option to be converted and processed in each line is used as input to the first *Select Response Option* (i.e., *Evaluate Response Options* and *Judge Response Options*). The remaining response options are discarded. Once a response is selected by each line of entities, nine copies are made and are transported using nine different links; three links associated with each of three lines of mechanisms to *Select Response Option* a second time (i.e., *Assign Tasks to Resources*). Again, the first response to be converted and processed in each line is used while the remaining responses are discarded. Finally, an executable response is generated in each line of *Select Response Option*, which marks the end of the Alternative #2 Arena® model. The portion of the Alternative #2 top-level functional flow which was modeled in Arena® is shown in Figure 4.

Clearly, the primary difference between the two alternative designs was the execution of the sub-functions of *Select Response Options*. Alternative #1, the Decentralized option, assigned one resource to *Evaluate Response Options*, *Judge Response Options*, and *Assign Tasks to Resources*, respectively. In contrast, Alternative #2, the Distributed option, assigned three resources in parallel to *Evaluate Response Options*, *Judge Response Options*, and *Assign Tasks to Resources*, respectively.

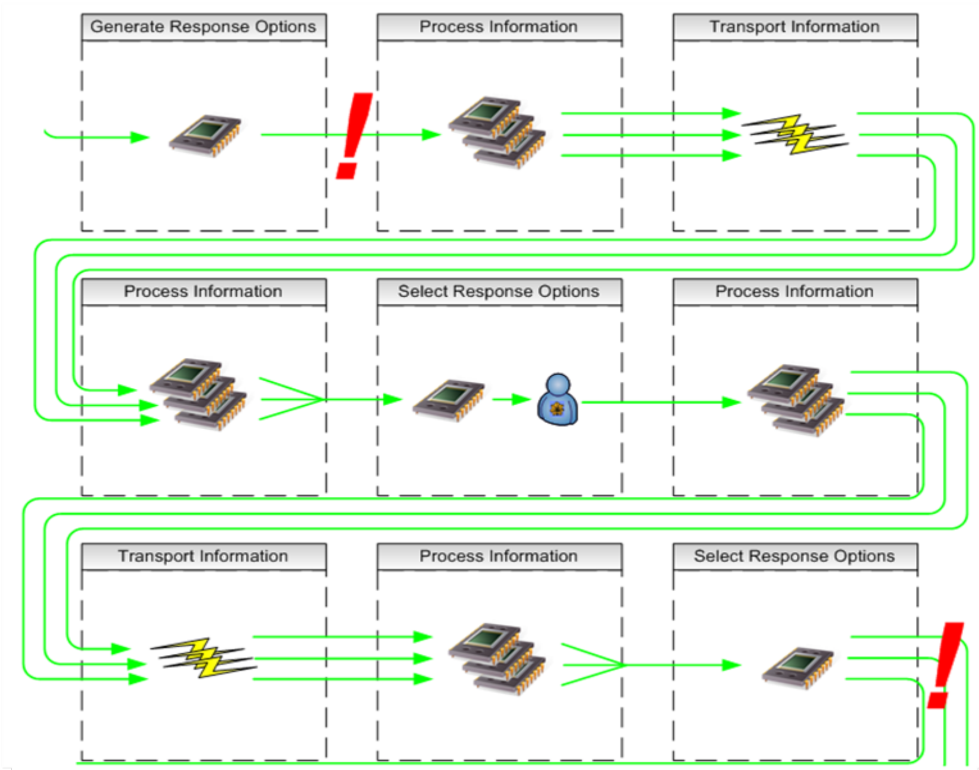

Figure 3: Alternative #1 - "Decentralized"

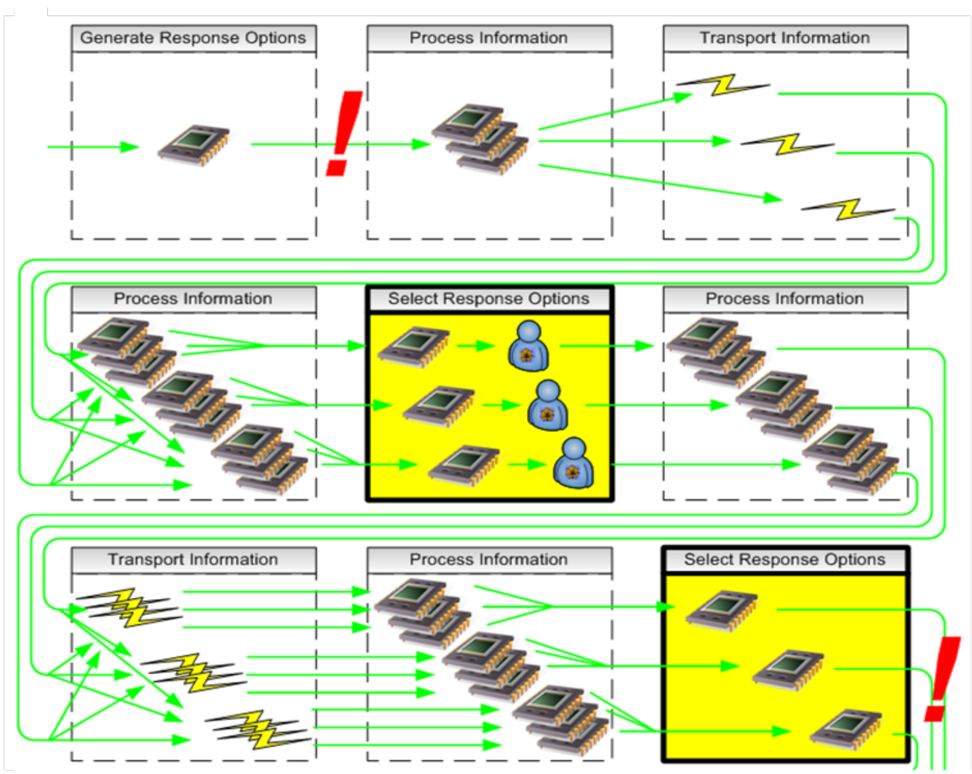

Figure 4: Alternative #2 - "Distributed"

4 THE SIMULATION

Performance of each alternative system design should be unique and it is the objective of discrete-event modeling and simulation to identify those differences. Differences in performance, as Buede (2000) states, are almost always related to an objective in the systems objective hierarchy and its related performance parameters. Given that a significant difference between the two alternative system designs was the change from decentralized to distributed decision authority and allocation authority, differences in performance of the alternatives should be apparent in the quality of response.

In particular, a combination of three Measures of Merit (MoMs), generated as part of the functional hierarchy, were selected to highlight differences in performance. First, the sum of time between response option being developed and response decision, and, time between order of response execution by decision-authorized entity and completion of allocations by allocation-authorized entity, was recorded for each alternative design. Second, the consistency of response between decision-authorized entities was recorded. These MoMs were selected, in part, because of the ability of Arena® to capture the statistics, but primarily because they should reflect the impact of distributed versus decentralized on the ability to make timely and correct decisions.

Thirty replications were conducted for each alternative model. Each replication began with a warm-up time of three simulation-minutes to fill queues and task resources followed by ten simulation-minutes in which data was collected. For both alternative models approximately 100+ response options were created and served as input during the ten operational simulation-minutes. The minimum time from response option creation until the generation of an associated executable response, the sum of the MoMs, was recorded for each response option and, in the case of Alternative #2, for each line of Select Response Option mechanisms. In addition, for Alternative #2, the executable responses for each response option were compared to determine consistency. An overview of the simulation results is presented in Table 1.

Table 1: Arena® Simulation Results - Overview

	Alternative #1	**Alternative #2**
Sample Mean of Minimum Time (sec)	16.313	14.955
Sample Standard Deviation of Minimum Time	1.382	2.630
Mean percentage of grouped Executable Responses which are consistent	100%	86.6%

Alternative #2 generates executable responses, on average, in less time than Alternative #1, but with more variance. In addition, Alternative #2 generates consistent executable responses approximately 87% of the time. Since Alternative #1 generates only one executable response for each response option, the consistency of response is 100% by default. Through hypothesis testing of the data collected, difference in the mean minimum time was determined to be statistically significant. This demonstrates the possible performance differences between the two alternative designs.

5 ISSUES

Specific conclusions on performance of the alternative system designs from the modeling and simulation results presented above should not be drawn. The objective of the modeling and simulation was not to conduct an analysis of design, but rather demonstrate the feasibility of using the developed architecture framework to conduct such analyses and compare alternative designs. Asserting performance superiority of one alternative over another should not be done for several reasons.

First and foremost, characteristics and attributes of the processes and resources were determined by the author and not by data collection or experimentation. For this reason statistically significant differences between designs apply only to the models developed and operationally significant differences cannot be assessed. Second, the modeling method used omitted many of the procedures and controls on the system identified in the architectural development. In other words, the models were a further abstraction of an already conceptual process.

Third, the Alternative #2 model developed was only one instance of the design solution space (i.e., the selection of three subordinated commanders and three processors in *Select Response Options*) possible from the operational architecture. Fourth, a value structure of value curves and weights for the systems objective hierarchy (Buede 2000), upon which trade-off decisions should be based, was not developed. Fifth, and finally, the above modeling and simulation represents only a few

MoMs for only one sub-task out of numerous tasks and missions required of a Surface Action Group (SAG) to secure local sea control.

6 CONCLUSIONS

Command and control (C2) is an enigma that has been studied by military leaders and warfare analysts for hundreds of years. As a result of the numerous definitions and concepts of C2 throughout history, the design of a C2 system is a daunting challenge to the systems engineer. Adding to the challenge is the understanding and integration of new operational concepts, such as Network-Centric Warfare, identified by stakeholders as necessary to meet their operational needs.

Research was conducted to dissect the complex engineering process of naval tactical C2 systems in order to identify the points of integration between doctrine and material. A survey of numerous historical texts and military publications was conducted to gain an understanding of various C2 concepts. Next, the system architecture methodology developed by Alexander Levis and presented by Buede (2000) and Levis and Wagenhals (2000), was followed. The process started with describing an operational concept, moved to the co-development of the functional architecture and the physical architecture, and concluded with the development of a notional operational architecture.

A portion of the contact prosecution process was modeled in the discrete-event simulation software Arena®. Simulations for both potential alternative designs were conducted. A comparison of results showed statistically significant differences in the measure MoMs, thus demonstrating the feasibility of the developed architecture framework to compare alternative designs.

REFERENCES

Alberts, D. S., Garstka, J. J., and F.P. Stein. 1999. *Network Centric Warfare: Developing and Leveraging Information Superiority* (2nd Ed.) [Electronic Version]. Washington, D. C.: Department of Defense Command and Control Research Program.

Alberts, D. S., and R. E. Hayes. 2006. *Understanding Command and Control* [Electronic Version]. Washington, D. C.: Department of Defense Command and Control Research Program.

Baran, P. 1964. *On Distributed Communications: I. Introduction to Distributed Communications Networks* [Electronic Version]. Santa Monica, California: The RAND Corporation.

Buede, D. M. 2000. *The Engineering Design of Systems: Models and Methods* [Electronic Version]. New York: John Wiley & Sons, Inc.

Cebrowski, A. K., and J. J. Garstka. 1998. Network-Centric Warfare: Its Origin and Future [Electronic Version]. *U.S. Naval Institute. Proceedings* 124(1), 28-35. Retrieved September 8, 2008, from http://proquest.umi.com.libproxy.nps.edu.

Levis, A. H., and L. W. Wagenhals. 2000. C4ISR Architectures: I. Developing a Process for C4ISR Architecture Design [Electronic Version]. *Systems Engineering* 11(3): 225-247.

AUTHOR BIOGRAPHIES

SHAUN P. HAYES is a Lieutenant, Engineering Duty Officer in the United States Navy and serves as an Assistant for Salvage at Naval Sea Systems Command, Washington Navy Yard. He received a MS in Operations Research from the University of Texas in 2003 and a MS in Systems Engineering from the Naval Postgraduate School in 2009. His email address is shaun.p.hayes@navy.mil.

EUGENE P. PAULO is an Associate Professor of Systems Engineering at the Naval Postgraduate School. He received a PhD in Industrial Engineering and Management Systems from the University of Central Florida in 1998. His research interests include modeling and analysis of combat systems, systems engineering, and system simulation. His email address is eppaulo@nps.edu.