Using MTWS For Human-in-the-Loop C2 Organizational Experiments[1]

LT Joan M. Wollenbecker, USN
C4I Academic Group
Naval Postgraduate School
589 Dyer Road
Monterey, CA 93943
(831) 656-2772
jmwollen@nps.navy.mil

William G. Kemple
C4I Academic Group
Naval Postgraduate School
589 Dyer Road
Monterey, CA 93943
(831) 656-3309
Kemple@nps.navy.mil

David L. Kleinman
C4I Academic Group
Naval Postgraduate School
589 Dyer Road
Monterey, CA 93943
(831) 656-3758
kleinman@nps.navy.mil

Gary Porter
C4I Academic Group
Naval Postgraduate School
589 Dyer Road
Monterey, CA 93943
grporter@nps.navy.mil

Susan Hocevar
Systems Management
Naval Postgraduate School
589 Dyer Road
Monterey, CA 93949
Shocevar@monterey.nps.navy.mil

[1] The office of Naval Research, Cognitive and Neural Sciences Division, A2C2 and the Institute for Joint Warfare Analysis (IJWA) at the Naval Postgraduate School supported this work.

Abstract

The Adaptive Architectures for Command and Control (A2C2) research project is examining issues in joint command and control, focusing on organizational adaptation. The project includes a series of "human-in-the-loop" experiments at the Naval Postgraduate School. The experiments are in three tiers, ranging from basic to applied/operational research, with tier 1 being the most basic. Four tier 1 experiments have been conducted to date, all employing the DDD-III simulator as the experimental driver. The DDD is designed for this type of research. It offers a high degree of control and supports on-line collection of data. It also involves a high level of abstraction, which is well suited to basic research. The basic A2C2 research will continue, but the research is also beginning to branch into the more operational/applied arena. The A2C2 team has selected the Marine Corps MTWS as the experimental driver for tier 2 experiments and has installed MTWS at NPS. The fifth A2C2 experiment used MTWS to reexamine selected research questions from experiment four, focusing on the performance of Joint Task Force decision-makers in model based and traditional JTF architectures. The architectures used in experiment five resemble as closely as possible those used in experiment four.

1.0 **Introduction**

The Office of Naval Research (ONR) sponsored Adaptive Architectures for Command and Control (A2C2) research team includes researchers from academia, industry and government. The team is examining issues in joint command and control, focusing on organizational adaptation, and ranging from basic to applied/operational research.

Previous research, with relatively simple organizations and tasks, has confirmed the common belief that organizational performance is higher when there is a "match" between the task and the organizational structure or architecture. Generalized, this leads to an underlying premise of the A2C2 research program, that joint organizational architectures should somehow match the available resources to the missions (task structures) at hand. Furthermore, changes to the resources available or the task structure should in turn induce changes to the joint organizational architecture. In order to design adaptable joint organizations, capable of effectively changing their architecture to maintain high performance, we need to increase our understanding of the interactions between joint organizational architectures and task structures. This is a key focus of the A2C2 program.

The A2C2 program uses a "design-model-test-model" (DMTM) framework to conduct its research. Within this framework, models are employed before the experiments ("tests") both to design organizational architectures and to pre-play the experiment. After the experiments models are compared to data collected during the experiment to validate and improve the models and to conduct additional analysis. The "test" or experimental portion of the research includes a three-tiered series of "human-in-the-loop" experiments at the Naval Postgraduate School (NPS). These experiments are designed in such a manner that data can be collected both to validate and improve the models and to examine the experimental research questions and hypotheses.

The three experimental tiers are associated with research at different levels of the basic-to-applied spectrum. Tier-1 experiments are intended to examine basic research questions. They tend to be narrow in scope with highly abstract scenarios, leading to limited realism. Tier-2 and -3 experiments are intended to help transition relevant findings into the more operational/applied arena. They are "envisioned to use more complex scenarios, be more operationally/strategically oriented, joint in nature, require longer trial times in order to capture events of interest, be more distributed in design, and finally, in tier-3, to make of use operating forces as subjects." (Porter, 1996)

Four tier-1 experiments have been conducted at NPS to date. All four have employed the Distributed Dynamic Decisionmaking III (DDD-III) simulator as the experimental driver (see Kleinman et al. 1996). This simulator is designed for, and ideally suited to, this type of basic research. It offers a high degree of experimental control and supports on-line collection of a variety of user-specified measures. It also involves a high level of abstraction, which is well suited to basic research.

All four tier-1 experiments have employed variants of the same basic scenario, which was abstracted from a joint training scenario used at the Armed Forces Staff College. Basically, a country friendly to the United States has been invaded by a neighboring state and has asked the United States for help. In response, a Joint Task Force (JTF) is tasked to conduct expeditionary amphibious operations to seize a seaport, an airport and a key bridge to facilitate the introduction of follow-on forces. The forces must accomplish a set of approximately 50 tasks, some known and some surprise, and some with temporal interdependencies, to achieve the overall mission. Developing concepts and methods to design architectures optimally matched to such a set of tasks, and comparing the performance of these architectures against that of more traditional architectures have been key foci of the A2C2 research. "Trigger" events that dramatically alter the task set or resources available have also been introduced during the scenario in two of the experiments (two and three) to examine structural and process adaptation.

The basic A2C2 research associated with tier-1 experiments (and the use of the DDD-III) will continue throughout the project, but the research is also beginning to branch into the more operational/applied arena, which involves tier-2 experiments. The A2C2 team has selected the Marine Air-Ground Task Force (MAGTF) Tactical Warfare Simulation (MTWS) as the experimental driver (see Porter, 1996) for tier-2 experiments, and with the support of ONR, MTWS has been installed in the NPS Systems Technology Battle Lab (STBL).

In order to effectively branch all areas of the research project into the more applied domain; a portion of the experimental research needs to smoothly transition to the tier-2 (MTWS) simulator. This led to the design of the fifth A2C2 experiment, which was conducted at NPS during February and March 1999, following the DMTM framework. Experiment five sought to reexamine a subset of the research questions examined in experiment four, but with the scenario played on the tier-2 experimental driver.

Each A2C2 experiment builds upon its predecessors, and experiment four, among other things, sought to clarify selected results of experiment three. Thus, both need to be discussed to set the stage for discussion of experiment five. This is done in the following section. Next, section 3 describes the design and conduct of experiment five, and section 4 presents some preliminary results and observations. Section 5 discusses several ancillary research questions and presents lessons learned, and section 6 discusses future work. The data extraction and manipulation discussions and analysis details are presented in the appendix. Full reports on the third experiment are presented in the 1998 CCRTS proceedings (see for example, Benson, et al. 1998; Curry et al. 1998; Entin et al. 1998; Handley, et al. 1998; Hocevar, 1998; Hocevar et al. 1998; Hutchins, 1998), and reports on the fourth are in this volume.

Four categories of data were collected during each of the experiments:

1. DDD simulator-collected data or MTWS data files
2. Video and audio tapes
3. Observer-collected data
4. Player self-report data

The results discussed in this paper are based on DDD, MTWS and observer-collected data.

2.0 **Experiments three and four**

Experiments three and four were conducted at NPS during November 1997 and August 1998 respectively, using architectures derived from modelers and operational experts. The DDD simulator was used to provide stimuli throughout the scenarios to student officers who performed as JTF and component commanders. Each player represented a commander, his or her staff and assigned forces. The following sections describe each experiment in more detail.

2.1 ***Experiment three***

The general theme of experiment three involved changes to the organization's structure induced by a sudden significant loss of resources.

2.1.1 ***Objectives of experiment three***

Several research areas were pursued in the third experiment. A primary focus was on the choice officers, playing the role of Joint Task Force DMs, made with respect to changes in the organization's structure. The objectives of the experiment were to:

- Gain knowledge about joint decision-making processes.

- Test the research hypotheses: Organizations will adapt to architecture *closer* to their current one, rather than to an optimal one that is *farther away*. (PROXIMITY vs. OPTIMALITY).

- Collect data to compare with model predictions (and thus improve the models). For this experiment, models generated architectures with certain inherent properties, predicted performance using those architectures before and after the trigger, calculated *distance* between architectures and simulated decision and coordination processes.

- Examine degree of coordination on tasks and organizational performance to determine whether organizations designed to require less coordination actually exhibited less, and whether this led to better performance.

2.1.2 *Results of experiment three*

To test the "choice" hypothesis in experiment three, players were well trained and familiarized in the original architecture, a "traditional" architecture that required a relatively high degree of coordination to accomplish the tasks. They then played in this architecture up to the insertion of a trigger event, which resulted in the loss of approximately 35% of the JTF's assets. Following the trigger, each team was given a choice of three architectures, all with the reduced asset set: the original six-node architecture they were familiar with; a five-node architecture relatively similar to the one they were familiar with, but designed to perform the mission better (including requiring less coordination); and a four-node architecture that was very different, but designed to perform the mission in an "optimal" manner. All teams decided to "stay" in their original architecture despite the heavy loss of assets. When asked to make a second choice, all but one team (eight of nine) chose the architecture that was relatively similar to the one they were familiar with. They then played the post-trigger scenario in two architectures, the one they chose and one other assigned in counterbalanced order. The architectures played in experiment three are depicted in several papers in last years CCRTS Proceedings, e.g. Benson, et al, 1998. The six and five-node architectures each used two communications nets, while the four-node architecture used a single net.

In terms of coordination, the architectures performed as designed. Based on a subset of the tasks selected before the experiment, the average number of nodes coordinating to accomplish each task was different between architectures, with organizations designed to require less coordination actually exhibiting less. Performance, on the other hand, was not as anticipated. The architecture that required the most coordination actually performed best. (Benson, et al, 1998) This unanticipated result contributed directly to the design of experiment four. Among other things, we wanted to determine whether the result was due to the teams being familiar with and trained in one architecture and not the other or the architectures themselves (e.g., design philosophy or the fact that the "optimal" architecture only had four nodes and thus four players while the traditional architecture had six).

2.2 *Experiment four*

The main focus of the fourth experiment was to investigate the factors leading to the results observed in experiment three.

2.2.1 *Objectives of experiment four*

While the primary goal of experiment four was to investigate the results of experiment three, several research areas were again pursued. The objectives were to:

- Gain knowledge about joint decision-making processes.
- Test the research hypotheses: For properly trained teams, model-based, optimized organizations will perform better than "traditional" ones.
- Collect data to compare with model predictions (and thus improve the models).

2.2.2 ***Architectures used in experiment four***[2]

Three architectures were again employed in experiment four, all with the reduced asset set from experiment three: a six-node traditional architecture that required a relatively high degree of coordination to accomplish the tasks (similar to the original architecture in experiment three), see figure 1 below; a six-node architecture designed to perform the mission in an "optimal" manner, see figure 2 below, and a four-node architecture that was also designed to perform the mission in an "optimal" manner (similar to the "optimal" architecture in experiment three), see figure 3 below. For experiment four, both six-node architectures employed two communications nets and the four-node architecture used a single net.

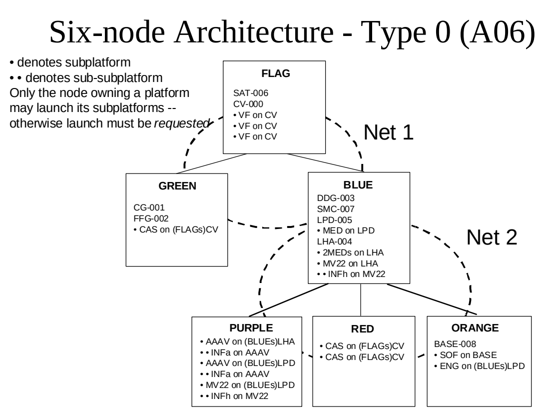

Figure 1. Six-node (A06) Architecture

---
[2] A2C2 Debrief from Experiment Four, August 1998, Entin, et al.

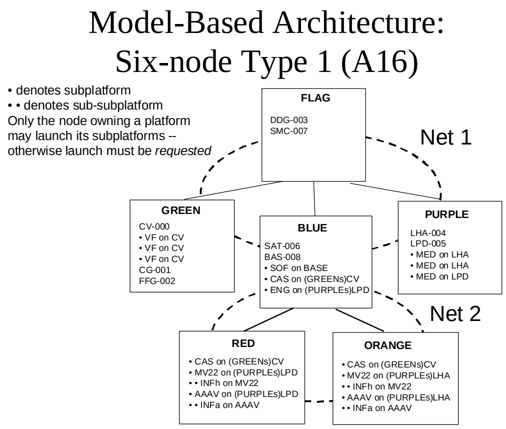

Figure 2. Six-node (A16) Architecture

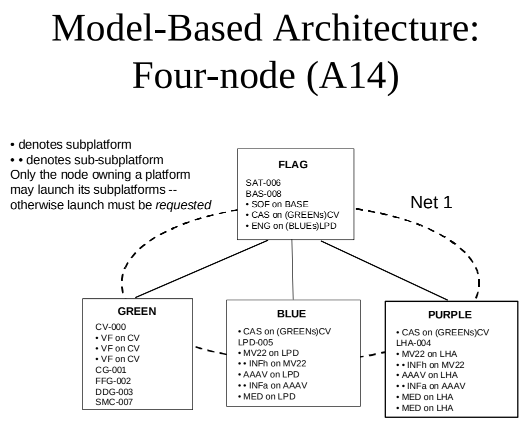

Figure 3. Four-node (A14) Architecture

2.2.3 ***Results of experiment four***[3]

The preliminary results from experiment four demonstrate that the model based A16 architecture outperformed the traditional A06 architecture significantly and also that the model based A14 architecture maintained performance compared to A06 despite a 33% manning reduction. More detailed results are given elsewhere in this volume.

Experiment five attempted to reproduce the second of the above findings – equivalent performance of the four-node, model-based architecture and the six-node, traditional architecture.

3.0 **Experiment five**

MTWS was used as the experimental driver for experiment five. The scenario and forces used, replicated as closely as possible those in the DDD and experiment four. The similarities and differences between DDD and MTWS will be discussed below. Like previous experiments, the

[3] A2C2 Debrief from Experiment Four, August 1998, Entin, et al.

scenario involved NPS officer students playing as JTF and component commanders. Again, each player represented a commander (DM), his or her staff and the assigned forces. The forces were utilized in either a four-node and one communications net (everyone can talk to each other) or a six-node and two communications nets (where one DM acts as the relay) architecture. The same basic architectures as were used in experiment four (A14 and A06); they were modified slightly as shown below in figures 4 and 5 to suit MTWS operations.

3.1 *Objectives of experiment five*

Experiment five is an attempt to use a tier II model – MTWS – in an abstract way to make it look like a tier I model – DDD – and compare the results of the two experiments to see how closely the two models reflect each other. Results of this comparison, based on observer and machine collected performance data are given in section 4 below.

In addition to examining whether the results of experiment four could be reproduced on MTWS, experiment five sought to examine the following.

- What are the similarities and differences in experimental control possible using MTWS?

- What is the feasibility (and resources required) to collect the same or similar measures using MTWS as collected from the DDD?

- What factors should be considered when selecting the experimental driver when the research question does not clearly favor one over the other (deciding between the DDD and MTWS)?

- In order to better control the initial comparisons between MTWS and DDD, MTWS was played in as highly an abstract mode as it was able to support (see Greenwood, 1998). What are the implications for using a high fidelity tactical simulator in a low fidelity environment?

- Does the use of trained operators between the DMs (subjects) and the experimental simulator affect performance?

- Will increased “Jointness” at lower levels in a JTF allow it to be comprised of fewer C2 nodes without adversely effecting performance given sufficient training of the DMs?

The results regarding the use of operators are given in section 4 below. The other questions are discussed in section 5, except the data extraction and manipulation discussions, which are in the appendix.

3.2 ***Architectures used in experiment five***

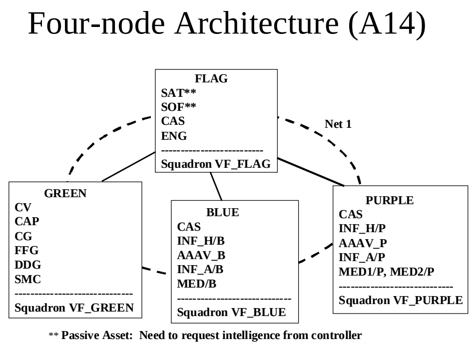

Figure 4. Four-node (A14) MTWS Architecture

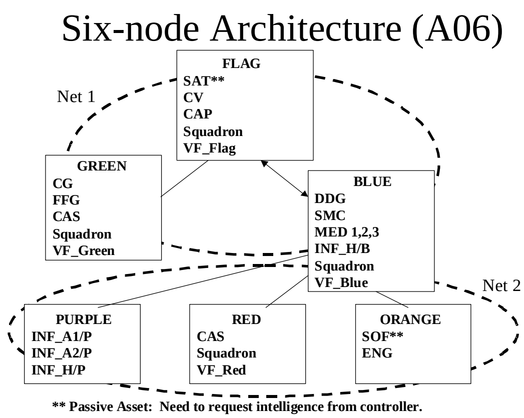

Figure 5. Six-node (A06) MTWS Architecture

Figure 4 shows the A14 architecture modified for MTWS (compare to figure 3). Similarly, for the A06 architecture, compare figure 5 to figure 1 above.

3.3 ***Conduct of experiment five***

The participants included 7 officer-students from the JC-81 “senior C4I systems” class and 12 officer-students from the JC-91 “junior C4I systems” class from NPS. The participants were divided into 5 four-node teams and 3 six-node teams. They participated in fourteen trials over a two-week period. Eight trials used the four-node A14 architecture (figure 4). To examine the possible requirement for trained operators, three of those had each single player act as both the DM and operator for a node, and five trials used pairs of players in each node. Six trials used the six-node A06 architecture (figure 5). All six-node trials used separate DMs and operators.

The JC-91 class was given 5 hours of general MTWS training and 5 hours of intensive operator training followed by 10 hours of operator practice. The JC-81 class had all been DMs and operators in experiment three using the DDD, and thus were familiar with the scenario, and would only perform as DMs during experiment five, so they were only provided with 5 hours of general MTWS training and 2 hours observing a JC-91 performance trial.

Each trial was scheduled for a two-hour block ending in a performance trial lasting 30 to 40 minutes. Teams playing a four-node or six-node architecture for the first time were given the first hour to adjust, using a training scenario and the same architecture.

4.0 **Results of experiment five**[4]

4.1 ***Replication of the DDD-III results of experiment four on MTWS***

Results based on both observer and machine collected data indicate that the experiment four finding is reproduced in experiment five – well trained, four-person teams playing in a model-based, optimized architecture perform as well as well trained, six-person teams playing in a "traditional" architecture.

Figure 6 shows the observer performance ratings for the two architectures. Note that there is no significant difference in observer performance rating between the four and six node architectures despite having two more people to complete the same tasks.

Observer Performance Ratings

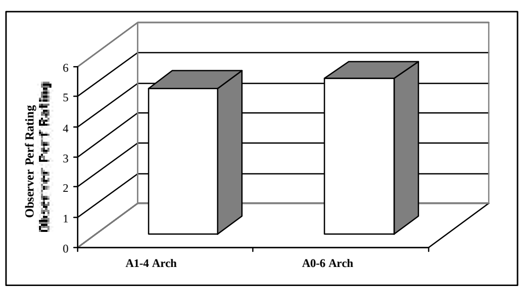

Figure 6. Observer Performance Ratings from experiment five.

Performance based on machine collected data is reflected in figure 7. It again shows no significant difference in performance between the four- and six-node architectures. In this case, the data are average accuracy scores reflecting how well the team brought the required assets to bear on a pre-selected set of five tasks. The data extraction and analysis are discussed in the appendix.

[4] Observer-based results from A2C2 Experiment Five Preliminary Results Debrief by Entin, et al., March 1999.

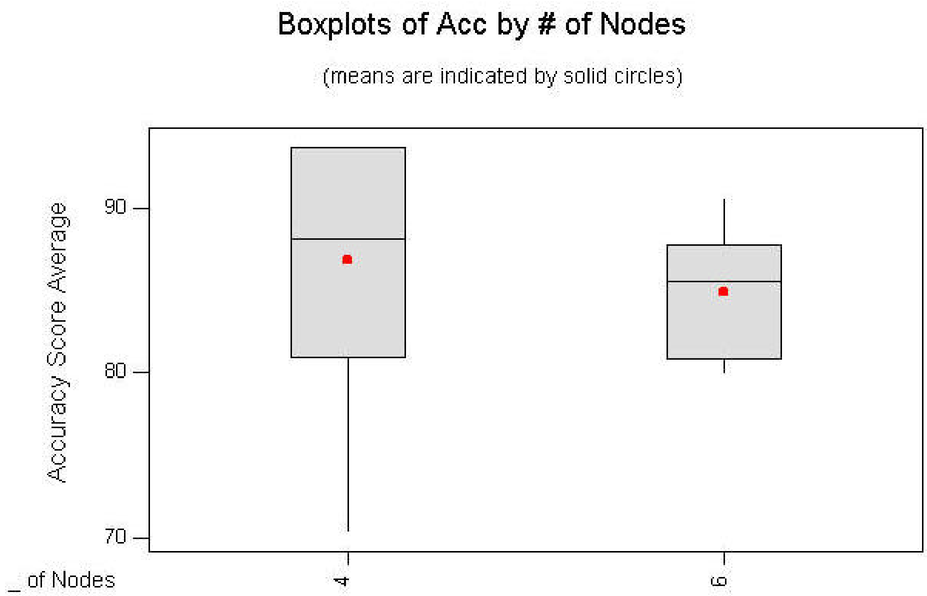

Figure 7. Boxplot of accuracy by # of nodes.

4.2 ***The effect of the use of trained operators between DMs (subjects) and the experimental simulator on performance***

To assess the effect of having separate operators and DMs in each node rather than having one person do both tasks (as was done on the DDD), three four-node runs were done with the DMs also being the operators. The differences in the observer performance rating are insignificant. But, there is an apparent anomaly, with higher observer performance ratings for teams where a single player acted as both DMs and operator. If it is real, it might be explained by two facts. First the JC-91 teams played as operators for other teams and as DMs with and without supporting operators for there own teams. Thus, they participated in a considerably higher number of trials. Second, the 3 trials in which JC-91 performed as both DMs and operators were the last three trials of the experiment. Both facts could indicate that learning contributed to the (not significant) differences in figure 8.

Operators vs. No Operators

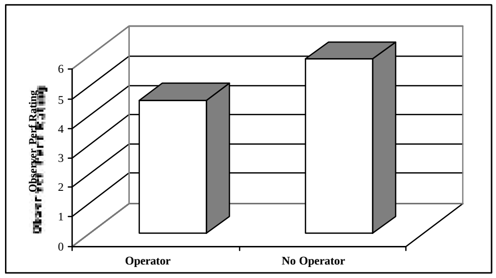

Figure 8. Operators vs. No Operators from experiment five.

4.3 *Workload breakdown*

Although it was not a specific research question, the workload distribution between nodes was compared across the architectures to see if the optimized architecture resulted in a more even distribution. The results based on observer data are shown in figures 9 and 10. It appears that workload was distributed more evenly across the nodes in the four-node architecture, but the differences are statistically insignificant. Of interest in both architectures is the workload associated with ground operations. In the four-node architecture (figure 4 above), purple and blue with the highest workload rating included the infantry units and conducted the ground operations. While in the six-node architecture (figure 5 above), the infantry and supportive CAS nodes (Purple and Red) had the highest workload.

Workload Across Architectures

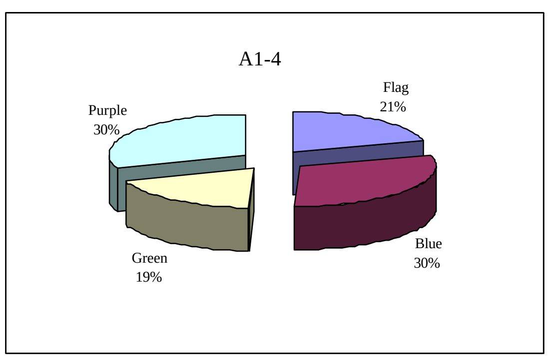

Figure 9. Workloads across architecture (A14) from experiment five.

A0-6 Workload

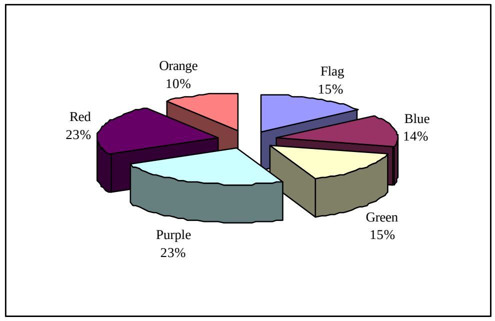

Figure 10. Workloads across architecture (A06) from experiment five.

Workload assessments based on MTWS data are shown for each team in figures 11 and 12. The workload was estimated by counting the number of commands issued by each player. The data parsing and Excel routines developed for handling the MTWS data are discussed in the appendix. Again, the distribution is not appreciably different.

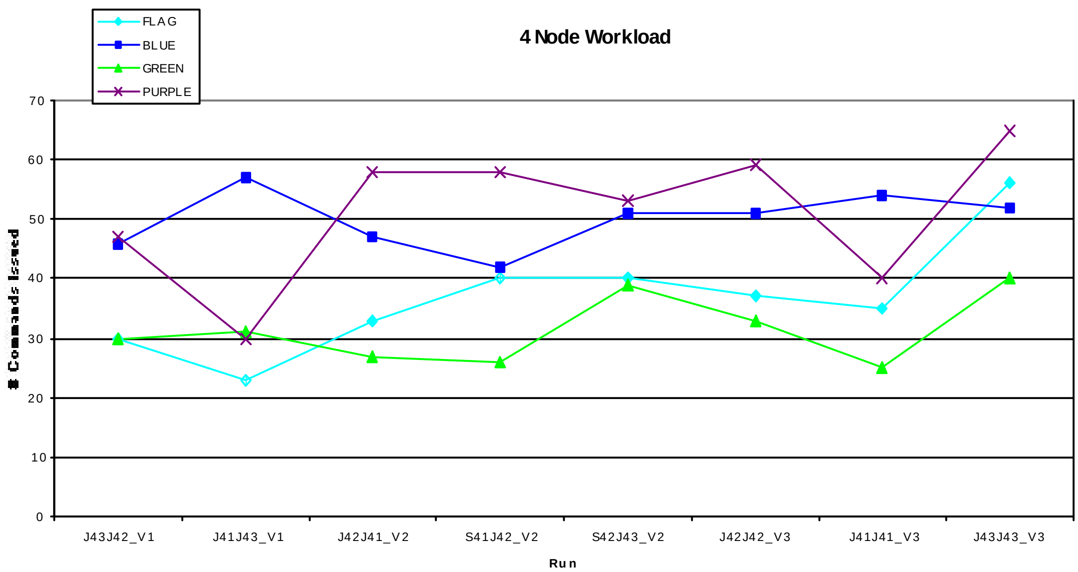

Figure 11. Four-node Workload from experiment five.

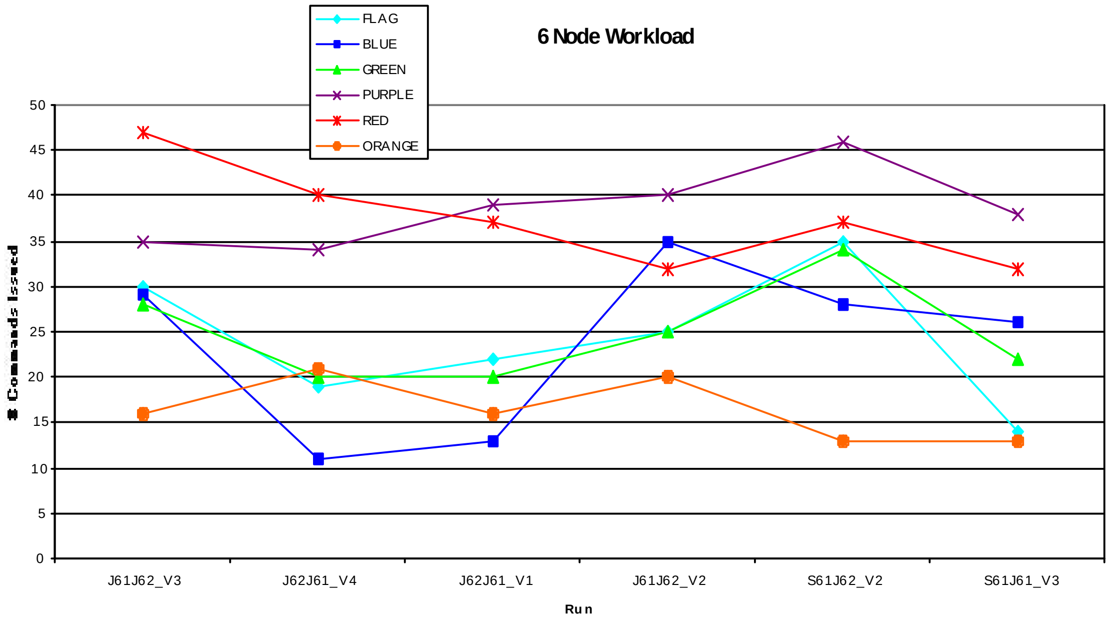

Figure 12. Six-node Workload from experiment five.

5.0 **Ancillary research questions and Lessons learned**

For convenience, the discussion of the use of operators was given in section 4 above, along with the results for the primary research question and an examination of workload distribution. The ancillary research questions are discussed in this section along with lessons learned observations. The data extraction and manipulation discussion is rather lengthy and detailed, so for ease of reading, it is presented in the appendix.

5.1 ***Differences and similarities in experimental control possible***

Recall that MTWS was configured to play like DDD, in an abstract mode providing ease of use at the expense of fidelity, in order to control the initial comparison between the two models. It was hoped that even though not all of the differences could be eliminated, enough of the more important ones could, resulting in outcomes from experiment five that matched experiment four. The amount of experimental control available to the researchers was different and the following sections discuss these differences.

5.1.1 ***Differences***

The Marine Corps database used in MTWS for this experiment used realistic parameters for movement speed, especially ground forces. As a result, the scenario could not be completed in the allotted 40 minutes. Rather than adjust the parametric database value for movement players were allowed “magic moves” in order to complete the scenario on time. After a task was completed ground units were effectively "beamed" to the next task location. The main differences between MTWS and DDD resulting from this were that all the teams completed the scenario, and most were well ahead of DDD completion times. Most teams complete their tasks in 15 to 25 minutes.

Another control difference was Red play, in DDD Red was scripted using software. In MTWS, parts of the Red forces were played by manually entering batch files or keyboard commands based on a written script. The Red player seemed to offer very consistent Red opposition for Blue and is not considered to be a factor. A sample Red script is shown in the appendix, figure A.11.

5.1.2 *Similarities*

The Common Operational Picture, architectures, tasks, resources, command structure, and in general the DOE issues remained the same in experiments four and five.

5.2 *The factors that should be considered when selecting the experimental driver*

The two key factors to be considered are what level of control versus what realism you want to incorporate into the experiment and how abstract you want to take a tactical simulator to better increase your control over the experiment.

5.2.1 *Control versus Realism*

There are always trade-offs to be considered when choosing a model for research. In this instance MTWS was chosen for its higher fidelity and for the fact that it is an established model that was developed to simulate similar scenarios to the one used in the A2C2. The trade-offs can be found in the following: abstractness vs. fidelity and control versus free-form human decision- making.

Realism has a downside in that it requires a complex interface to create and play the scenarios. The DDD has the advantage of ease of creating scenarios. Time is also a critical factor in choosing a model – time to run the scenario, train the players, and obtain the necessary data in as little a time as possible. Time is also a factor within the scenario on the model – is there time to react to arcade style stimuli versus time in the run to think and plan.

Realism in experiment five also necessitated additional players to complete the scenario, so the complexity of the interface and data displayed is also a critical factor. Numbers of support players to build and maintain a database and to construct and change scenarios can be critical if there is not enough personnel or time for development. Numbers of supporting players to perform control and OPFOR tasks proved to be difficult with MTWS. Experiment five used enlisted petty officers as the OPFOR and to run the overall scenario in the STBL. The number of analysts and time to conduct post trial data analysis was another consideration.

Ability to support necessary communications and the ability to record such communications for post game analysis must also be considered.

5.2.2 *Implications for using a tactical simulator in a highly abstract mode*

One of the key problems is just getting MTWS to be as abstract as possible. MTWS throughout the weeks leading up to the actual trial would inject random variables into each game, even though the scenario was scripted. Just to see what the submarines in the game could do – they had been designed with no weapons – one was allowed to get within striking distance of the carrier and it "took it out" by simply engaging it, with the result that no more aircraft could be launched for the rest of the game.

Using MTWS with both operators and DMs combined with player time constraints meant that only a small number of trials were possible to reduce any learning effects over time. The more realistic the model the smaller the number of trials and smaller amount of data possible. The more abstract and strictly controlled the model, the larger the number of trials possible – less training and no separate operators are required.

MTWS throughout the trials had stochastic traits that had to be expected. The key one seen in all the trials was the interaction of the engineers and the minefields on both roads. In some cases the mine fields could be removed with no casualties and at other times it would inflict heavy casualties – which necessitated the Blue forces being larger than needed to complete the tasks.

5.3 ***Does the use of trained operators between DMs (subjects) and the experimental simulator affect performance?***

Providing an operator cuts down on the DM workload. Using, DDD operators were not required due to the relative ease of operator inputs. However, since MTWS is a higher resolution model that provides much more detailed information to the player, lack of an operator could have impacted a team's performance. A word of caution however, trained operators can be overly effective and can bias the results by providing more assistance during a trial than intended. Performance increases may be due to operator performance instead of DM performance.

5.4 ***Will increased "Jointness" at lower levels in a JTF allow it to be comprised of fewer C2 nodes without adversely effecting performance, given sufficient training of the DMs?***

Based on some previous work, it was anticipated that the model-based architecture would be considerably more "joint" (encompass assets in more warfighting domains) in the lower nodes than the traditional architecture. This did not turn out to be the case. It cannot be reported on at this time. This should be followed-up in future experiments.

6.0 **Conclusions**

As stated, the primary focus of experiment five was to determine whether the performance results of experiment four (equivalent performances of the four-node, model-based architecture and the six-node, traditional architecture) could be reproduced on the higher fidelity MTWS simulator. This result was reproduced, using both the observer assessments and the simulator collected data.

In experiment three, the coordination required to complete the selected tasks was examined and found to match expectations, the model-based architecture required significantly less coordination than the traditional architecture. This was also examined for experiment five using the simulator collected data (see the appendix) and the result was reproduced. Players had to coordinate more in the traditional architecture

Based on these results, we believe that we can employ whichever simulator is appropriate for the research question at hand without fear that results will be confounded with choice of simulator. The characteristics of the architectures tend to manifest themselves regardless of simulator. Several things that should be considered in selecting the simulator for any given experiment are discussed in section 5. We can consider these in their own right without undue concern of adverse effects on the experimental outcome.

7.0 **References**

Benson, R., Kemple, W.G., Kleinman, D., Porter, G., Serfaty, D. (1998) *An example of model-based empirical research: A soup-to-nuts evaluation of alternative C2 architectures*. Proceedings for the 1998 Command & Control Research and Technology Symposium (pp. 34-43). Monterey, CA, June 1998.

Curry, M.L., Kleinman, D.L., & Pattipati, K.R. (1998) *On mission planning: A foundation for organizational design*. Proceedings for the 1998 Command & Control Research and Technology Symposium (pp. 1-10). Monterey, CA, June 1998.

Entin, E., Serfaty, D. & Kerrigan, C. (1998) *Choice and performance under three command and control architectures*. Proceedings for the 1998 Command & Control Research and Technology Symposium (pp. 132-137). Monterey, CA, June 1998.

Greenwood, D.F. (1998) *Adapting the A2C2 Experiment for use with MTWS*. Masters Thesis, Naval Postgraduate School, June 1998.

Handley, H.A.H., Zaidi, Z.R. & Levis, A.H. (1998*) Pre-experimental modeling of adaptive organizational architectures*. Proceedings for the 1998 Command & Control Research and Technology Symposium (pp. 44-53). Monterey, CA, June 1998.

Hocevar, S.P. (1998*) Deciding to adapt organizational architecture: Facilitators and inhibitors to change*. Proceedings for the 1998 Command & Control Research and Technology Symposium (pp. 78-95). Monterey, CA, June 1998.

Hocevar, S.P., Kemple, W.G. & Benson, R. (1998) *Translating simulation-based findings to an operational arena: Interpretations from research participant-experts*. Proceedings for the 1998 Command & Control Research and Technology Symposium (pp. 54-65). Monterey, CA, June 1998.

Kleinman, D. L., Young, P. W., Higgens, Gregory S., (1996) *The DDD-III: A tool for Empirical Research in Adaptive Organizations*. Proceedings for the 1996 Command & Control Research and Technology Symposium (pp. 827-836). Monterey, CA, June 1998

Porter, G. R., (1996) *Joint Simulation Assessment*. Proceedings for the 1996 Command & Control Research and Technology Symposium (pp. 819-826). Monterey, CA, June 1996.

APPENDIX

A.0 **The feasibility (and resources required) to collect the same or similar measures**

DDD automatically collects data within its software and can be relatively easily customized for data collection. MTWS sends predetermined types of game play information to files. The two files that are important for data collection in MTWS after a run are *Spot_Report_Log* and *command.history*.

A *Spot_Report_Log* contains a record of all spot reports received by the workstation. A spot report is generated whenever an operator enters data and any corresponding actions resulting from that data entry. Sample reports are listed below. The *Spot_Report_Log* gathers all the spot reports generated during a game. The contents of a *Spot_Report_Log* are dependent on what controllers were selected for the workstation (they varied from MANCON 1 to 7); to generate a comprehensive log for the game, the red or opposition force controller station (MANCON 7); was set to collect all spot reports for the game. This is not without risk, however, in one run of the scenario the red station was not set and data was collected from each of the workstations individually. Collecting the data from an individual workstation requires that the researcher must time correlate all the events instead of all the events being in one file and time correlated by the computer. The terminals used were Hewlett Packard Unix workstations, set up in their own local area network. There were a total of nine workstations.

The *command.history* file contains every valid command entered at any of the participating workstations. Extracting the data was accomplished by data parsing algorithms developed by LT Ron Soule, USN, written in Turbo Pascal 4.0. The data was formatted for direct input into MS Excel for final analysis.

Samples of various reports received are listed below followed by selected portions of the Turbo Pascal program.

A.1 **Samples of various reports received**

Air Report. The cruise missile launched by MANCON 2 has reached attack point. The corresponding status change report for SUB2 to AGCON1 stating that it is under air attack at the UTM coordinates 32SQF124981.

;AIR RTE POINT;ASW3;32SQF124981;060814ZJAN99;MANCON_2;CRUISE MISSILE HAS REACHED ATTACK POINT
PRINTED BY: mds007

;STATUS CHANGE;SUB2;32SQF124981;060814ZJAN99;AGCON_1; UNDER AIR ATTACK
PRINTED BY: mds007

Detection Report. Silk 2 is reporting to AGCON 1 that it has detected a structure at UTM coordinates 32SPF879851.

;OBJECT DETECT;SILK2;32SPF866877;060814ZJAN99;AGCON_1;HAVE DETECTED STRUCTURE AT 32SPF879851;
PRINTED BY: mds007

Generic assessment report. The assessment report to the FFG that their air-to-surface attack occurred, however since no one was there, who was under MANCON 2's control, there was no assessment of the attack.

;ASSESSMENTREPORT;FFG;AIR_TO_SURFACE;32SQF124981;060814ZJAN99;
MANCON_2;
NO ASSESSMENT; Msn# ASW3
PRINTED BY: mds007

Ground engagement report.

;REPORT; GROUND_ENGAGEMENT2; 32SPF844929; 060814ZJAN99; MANCON_4, AGCON_1;
Initial engagement times:
INF_A/P; 060806ZJAN99
HILLIFY; 060806ZJAN99
INF_H/P; 060809ZJAN99
VMV_PURPLE; 060810ZJAN99
Cumulative losses:
UNIT HILLIFY; TROOPS 5 WIA, 1 KIA UNIT AGSUP; No damage assessed. UNIT AGSUP; No damage assessed.
UNIT HILLIFY; HMMWV-40MM-MG 1 K_KILLED,
UNIT INF_H/P; TROOPS 2 WIA,
UNIT VMV_PURPLE; TROOPS 4 WIA, UNIT AGSUP; No damage assessed.
UNIT FROG2; TROOPS 3 WIA,
Incremental losses (increases since the last report)
UNIT HILLIFY; TROOPS 3 WIA, 1 KIA
UNIT VMV_PURPLE; TROOPS 4 WIA,
UNIT INF_H/P; TROOPS 2 WIA,
UNIT HILLIFY; HMMWV-40MM-MG 1 K_KILLED,
PRINTED BY: mds007

The ground engagement report shows the detail given by MTWS. INF_A/P and INF_H/P are infantry units controlled by the player named “purple.” HILLIFY is the corresponding red infantry unit being attacked. VMV_Purple are the MV-22's used to transport the purple infantry.

A.2 **Extracting and manipulating the data**

A.2.1 ***The beginning of the Turbo Pascal program written to parse the MTWS data follows (figure A.1)***

```
PROGRAM MTWSParser;

USES
   crt,
   dos;

VAR
   infile,
   parsefile,
   dumpfile   : text;

   PROCEDURE OpenFiles (VAR infile, outfile, dumpfile : text);

   VAR
      ok            : boolean;
      infilename,
      outfilename,
      dumpfilename  : string;

      FUNCTION ValidFileName (filename: string) : boolean;

      VAR
         f : text;   {text file}

      BEGIN {function ValidFileName}
         assign (f,filename);   {associates a file name with a file pointer}
         {$I-}                  {turns off I/O Checking to prevent an error
                                 from crashing the system}
         reset (f);             {resets the file pointer}
         {$I+}                  {turns on I/O Checking and obtains an
                                 IOResult}
         IF (IOResult = 0) THEN {an IOResult of 0 indicates no problems}
            ValidFileName := true
         ELSE                   {any other IOResult indicates a problem in
                                 resetting the file pointer, e.g., the
                                 filename was not invalid}
            ValidFileName := false;
      END; {funciton ValidFileName}

   BEGIN {procedure OpenFiles}
      ok := false;
      WHILE NOT(Ok) DO
      BEGIN {while loop}
         write ('Enter the name of the input file:  ');
         readln (infilename);
         Ok := ValidFileName (infilename);
      END; {while loop}
      assign (infile, infilename);
      reset (infile);
      write ('Enter the name of the output file:  ');
      readln (outfilename);
      assign (outfile, outfilename);
```

```
      rewrite (outfile);
      write (outfile,';DTG;UID;UTM;CTLR;REPORT;MESSAGE;TRACK');
      write (outfile,';UID2;UTM2;STATUS;TYPE;# UNITS;WIA;KIA');
      writeln (outfile,';K_KILLED;M_KILLED;F_KILLED;UNDAMAGED;NON MISS
CAPABLE;DESTROYED;DTG2');
      dumpfilename := 'dumpfile.txt';
      assign (dumpfile, dumpfilename);
      rewrite (dumpfile);
   END; {procedure OpenFiles}

   PROCEDURE ParseTheFile (VAR infile, outfile, dumpfile : text);
```

Figure A.1. Sample of the Turbo Pascal Data Parsing Program

A.2.2 ***Main Program***

Figure A.2 below shows how the main program works. The parser program was used on the Unix workstations.

```
BEGIN {main program}
      clrscr;
      OpenFiles (InFile, ParseFile, DumpFile);
      ParseTheFile (InFile, ParseFile, DumpFile);
      CloseFiles (InFile, ParseFile, DumpFile);
END. {main program}
```

Figure A.2. Data Parsing Program.

A.2.3 ***The Parser***

Figure A.3 shows how the parser works specifically. The results of the parser are the *Spot_Report_Log* and *command.history* file shown above in the sample reports and in figure A.4.

PROCEDURE ParseTheFile

WHILE NOT(eof(inFile)) DO
- get a line of text from the input file
- get the key (type of report)
  field := GetField(s);
- Processes the 51 types of reports in the following categories:
  - Air Operations
  - Ground Operations
  - Ship Operations
  - CE Operations
  - Fire Missions
  - Assessment Reports
  - Miscellaneous Status Reports

Figure A.3. How the parser works

A.2.4 ***The Excel spreadsheet***

After Parsing, the data was FTP'd to a PC, put into an Excel spreadsheet and sorted. A sample of the sorted data in Excel is shown in figure A.4.

DTG	UID	UTM	CTLR	REPORT	MESSAGE
060800ZJAN99	SOF	32SPG726000	MANCON 1	OBJECT DETECT	HAVE DETECTED STRUCTURE
060800ZJAN99	SOF	32SPG726000	MANCON 1	OBJECT DETECT	HAVE DETECTED BRIDGE
060800ZJAN99	SOF	32SPG726000	MANCON 1	OBJECT DETECT	HAVE DETECTED BRIDGE
060800ZJAN99	SOF	32SPG726000	MANCON 1	OBJECT DETECT	HAVE DETECTED RIVER
060801ZJAN99	H1	32SQG099006	MANCON 2	AIR RTE POINT	CRUISE MISSILE HAS REACHED ATTACK POINT
060801ZJAN99	FFG	32SQG099007	MANCON 2	ASSESSMENT REPORT	AIR_TO_SURFACE Msn# H1
060801ZJAN99	SOFFCAS1	32SPG901122	MANCON 1	AIR MSN LAUNCH	AIR MISSION HAS LAUNCHED
060801ZJAN99	FCAS1	32SPG901122	MANCON 1	AIR MSN LAUNCH	AIR MISSION HAS LAUNCHED
060801ZJAN99	GCAP1	32SPG901122	MANCON 2	AIR MSN LAUNCH	AIR MISSION HAS LAUNCHED
060801ZJAN99	AA86	32SPF625617	AGCON 1	AIR MSN LAUNCH	AIR MISSION HAS LAUNCHED
060801ZJAN99	LHA	32SPF962990	MANCON 4	AIR TRK DATA	AIR TRACK DETECTED
060801ZJAN99	AA86	32SPF658621	AGCON 1	AIR MSN ILLUMINATED	AIR MISSION ILLUMINATED BY NON-SAME SIDE RADAR
060801ZJAN99	LHA	32SPF962990	MANCON 4	AIR TRK DATA	AIR TRACK DETECTED
060801ZJAN99	LHA	32SPF962990	MANCON 4	AIR TRK DATA	AIR TRACK DETECTED
060801ZJAN99	LHA	32SPF962990	MANCON 4	AIR TRK DATA	AIR TRACK DETECTED
060801ZJAN99	GCAP2	32SPG901122	MANCON 2	AIR MSN LAUNCH	AIR MISSION HAS LAUNCHED
060801ZJAN99	FCAS2	32SPG901122	MANCON 1	AIR MSN LAUNCH	AIR MISSION HAS LAUNCHED
060802ZJAN99	BCAS1	32SPG901122	MANCON 3	AIR MSN LAUNCH	AIR MISSION HAS LAUNCHED
060802ZJAN99	SOFFCAS1	32SQG025193	MANCON 1	AIR RTE POINT	AIR MISSION HAS REACHED ORBIT POINT

Figure A.4. Sample of the parsed data in Excel.

There are additional 12 columns to each set of parsed data and the report average over 1100 lines of data collected for each run.

A.3 **DDD III and MTWS data**

The type of data collected by both games is different. DDD automatically scores the games based on a number of factors MTWS does not. The five factors collected from MTWS via the parser and Excel measured during post game analysis sessions were manually extracted from the parsed data and manually scored. The scorings were hand tabulated from the above sample data files. Routines will be written to automate this as appropriate.

A.4 **Accuracy scores and analysis**

Table A.1 lists the accuracy score (averaged across all five preselected tasks) for each experimental trial.

Run #	Date	Decision Maker	Operator	Variant	# of Nodes	Accuracy Score Avg
1	2/23/99	J41	J43	V1	4	88.15%
2	2/24/99	J43	J42	V1	4	70.30%
3	2/24/99	J42	J41	V2	4	88.15%
4	2/25/99	S41	J42	V2	4	93.75%
5	2/25/99	S42	J43	V2	4	93.75%
6	2/26/99	J61	J62	V3	6	79.96%
7	2/26/99	J62	J61	V4	6	81.22%
8	3/1/99	J62	J61	V1	6	86.82%
9	3/1/99	J61	J62	V2	6	86.82%
10	3/2/99	S61	J62	V2	6	90.63%
11	3/2/99	S61	J61	V3	6	84.34%
12	3/3/99	J42	J42	V3	4	78.58%
13	3/3/99	J41	J41	V3	4	88.15%
14	3/3/99	J43	J43	V3	4	93.75%

Table A.1. Excel spreadsheet showing accuracy score data from all runs of experiment five.

The accuracy scores were analyzed using Minitab One Way Analysis of Variance with the number of nodes as the factor. The differences were insignificant (P=0.626). This compares favorably with the results of the accuracy in experiment four (compare table A.1 to paragraph 2.2.3 above). Due to the small sample size the experiment does not prove conclusively that the DDD results could be replicated on MTWS, but this is a good step towards that goal.

Similar in both experiments were the data collected manually by observers during game play. The next version of MTWS contains an After Action Review (AAR) system. This will help automate and customize data collection capabilities of MTWS. Accuracy for experiment five was calculated using an excel spreadsheet designed by MAJ Jon Cook, USMC. Figure A.5 shows an example accuracy calculation for North Beach.

N Beach	Unit	Qty	AAW Px	ASUW Px	ASW Px	GASLT Px	FIRES Px	ARMR Px	HOLD Px	MINE Px	MED Px
	INF	2	2	0	0	20	4	4	20	2	0
	CAS	1	1	3	0	0	10	8	0	1	0
	DDG	0	0	0	0	0	0	0	0	0	0
	ENG	0	0	0	0	0	0	0	0	0	0
		ForceTotal, r =	3	3	0	20	14	12	20	3	0
		Required Power, R_i =	0	0	0	10	14	12	0	0	0
		Adjusted r_i =	na	na	na	10	14	12	na	na	na
		r_i/R_i =	0	0	0	1	1	1	0	0	0
		$[r_i/R_i]^2$ =	0	0	0	1	1	1	0	0	0
		$\Sigma [r_i/R_i]^2$ =	3								
		$(1/n)^*\Sigma [r_i/R_i]^2$ =	1	=Accuracy							

Nonzeros, n =	3

Resource Capabilities

Asset	AAW	ASUW	ASW	GASLT	FIRES	ARMR	HOLD	MINE	MED
CAS	1	3	0	0	10	8	0	1	0
DDG	10	10	1	0	9	5	0	0	0
ENG	0	0	0	2	0	0	2	5	0
INF	1	0	0	10	2	2	10	1	0

Task Requirements

Hill	0	0	0	0	14	12	0	0	0
Beach	0	0	0	10	14	12	0	0	0
Airport	0	0	0	20	10	4	0	0	0
Seaport	0	0	0	20	10	4	0	0	0
Bridge	0	0	0	8	6	0	0	4	0

Figure A.5. Accuracy experiment five.[5]

Accuracy is based on the resource capabilities (1st table in figure A.5) of the assets used and what the task required (2nd table in figure A.5). The accuracy for the north beach in this case was 1. The accuracy was then brought to the overall accuracy chart and multiplied by 100, so that for this task the accuracy was 100%. This is shown in Figure A.6 below.

	Assets Used					
Objective	**CAS**	**SHIP**	**INF**	**ENG**	**# of Controllers**	**Accuracy Score**
North Beach	BCAS3		INF_H/B INF_A/B		2	100.00%
Hill	PCAS1 PCAS2 PCAS3		INF_H/B INF_A/B INF_A/P INF_H/P		2	100.00%
Airport	BCAS2		INF_H/B INF_A/B		1	100.00%
Seaport			INF_A/P INF_H/B		2	72.00%
Bridge	PCAS6			ENG	2	68.75%
					Avg	88.15%

Figure A.6. Accuracy score for one run on all objectives.

[5] MAJ Jon Cook, USMC designed the Excel spreadsheet for the accuracy scores data analysis. The Resource Capabilities and Task Requirements tables are from Entin, et al., 1996.

The assets used in the columns in figure A.6 were hand counted from the data spreadsheets shown in figure A.4. Each of the tasks had a specific requirement to use certain assets to accomplish the task. Table A.2 below shows what assets were needed to complete each task. The DMs had a copy of this information for completing the mission.

Tasks		**Suitable Force Packages (See Note 1)**		
Task Name	Value	Option 1	Option 2	Notes
Take hill	20	1 CAS + DDG + 1 INF	2 CAS + 1 INF	See Note 2
Take Beach	20	1 CAS + DDG + 1 INF	2 CAS + 1 INF	See Note 2
Airport	30	1 CAS + 2 INF		See Note 2
Seaport	30	1 CAS + 2 INF	(DDG or CG) + 2 INF	See Note 2
Hold Hill	10	1 INF (must leave on hill)		
Lead vehicle	**15**	**1 CAS with intel via SOF**		See Note 3
Bridge	15	1 CAS + ENG		See Note 2
Medivac (See Note 4)	5	MED		
Sea Mines	10	SMC		
Artillery	2	DDG	1 CAS	
Frogs	10	DDG	1 CAS	
Ground Mines	5	ENG		
Tank	5	2 CAS		
Silkworm	**15**	**1 CAS with intel via SAT/SOF**		See Note 3
SAM Site	**10**	**1 CAS with intel via SAT/SOF**		See Note 3
Hind Helo	4	1 CAP	CG or DDG	
Hostile Aircraft	15	1 CAP	CG or DDG	
Submarine	25	FFG	DDG	
Patrol Boat	**15**	**1 CAS with intel via SAT**	**CG or DDG or FFG**	See Note 3

Note 1: Any other force packages will result in casualties or overkill/waste
Note 2: Stand-off INF or ENG when doing combined attacks with CAS or DDG
Note 3: Items in **BOLD** need to be positively ID'd (vs. neutrals or decoys).
Note 4: Attacks on most ground targets have possible casualty consequences that may require medivac. Medivac tasks have a short time window - 5 mins in which to accomplish.

Table A.2. Assets needed to complete each task and options.

Table A.3 below lists the types of assets available, their capabilities and the symbol used to identify them. DMs were also provided with this list. Each individual asset was labeled with the first letter of their position; for example the first Flag CAP launched would be FCAP_1. The operators were responsible for labeling their individual air assets, the naming convention used above was decided on for uniformity throughout the runs.

Assets/Platforms

Type*	Asset Name	Symbol	Capabilities
S	Detroyer	DDG	
S	Frigate	FFG	
S	Cruiser	CG	
S	Aircraft Carrier	CV	Has CAP, CAS
S	Landing Ship	LHA	Has MV22, MED
S	Landing Ship	LPD	Has MV22, MED, ENG
A	Engineers	ENG	launch from LPD
G	Infantry	INF_A (note 2)	launch from AAAV; confined to roads
A	Close Air	CAS	launch from CV
A	Fighters	CAP	launch from CV
A	Medivac (note 1)	MED	launch from LHA and/or LPD
S	Mine Sweeper	SMC	
S	Beach lander	AAAV	
A	Troop Helo	MV22	launched from LHA and/or LPD
A	Satellite	SAT	call controller for intel info
G	Special Ops	SOF	call controller for intel info
G	Infantry	INF_H	assumed to be on MV-22's

Note 1: MED once launch have < 5 mins to complete their mission
Note 2: Must locate 1 company of INF_A (from AAAV's) at each beach; then move them inland on roads.
*Type S = Surface Asset
Type A = Air Asset
Type G = Ground Asset

Table A.3. Asset capabilities and type. Also lists symbol used for identification.

A.5 **An automated method of collecting MTWS data from the parsed Excel spreadsheets**

Figure A.7 is a spreadsheet created in Excel to help automate the task of collecting data from the parsed Excel spreadsheets. Maj Jon Cook, USMC developed it after the manual counting had been completed. Figure A.8 shows a map created in Excel to help automate the task of collecting data from the parsed Excel spreadsheets. The map goes with the spreadsheet in figure A.7 that requires you to enter the UTM coordinates which then tells you if the coordinates are in any of the areas of interest or if it is outside those coordinates.

Label	UTM Grid	E	N		Modify	x	y		Objective		Hill	N Bh	S Bh	N Bdg	S Bdg	Port	Arprt	Location
Hill	32SPF835929	835	929	F	no	835	929		**Hill**		Yes	0	0	0	0	0	0	**Hill**
N Bch	32SPF845938	845	938	F	no	845	938		**N Beach**		0	Yes	0	0	0	0	0	**N Beach**
S Bch	32SPF861891	861	891	F	no	861	891		**S Beach**		0	0	Yes	0	0	0	0	**S Beach**
N Brdg	32SPG737007	737	007	G	yes	737	1007		**N Bridge**		0	0	0	Yes	0	0	0	**N Bridge**
S Brdg	32SPF734994	734	994	F	no	734	994		**S Bridge**		0	0	0	0	Yes	0	0	**S Bridge**
Seaport	32SPG814054	814	054	G	yes	814	1054		**Port**		0	0	0	0	0	Yes	0	**Port**
Airport	32SPF714842	714	842	F	no	714	842		**Airport**		0	0	0	0	0	0	Yes	**Airport**
Sea	32SPF865995	865	995	F	no	865	995		**Undetermined**		0	0	0	0	0	0	0	**Ocean**
Land	32SPF800900	800	900	F	no	800	900		**Undetermined**		0	0	0	0	0	0	0	**Land**

Figure A.7. Spreadsheet used in conjunction with map for UTM coordinate determination.

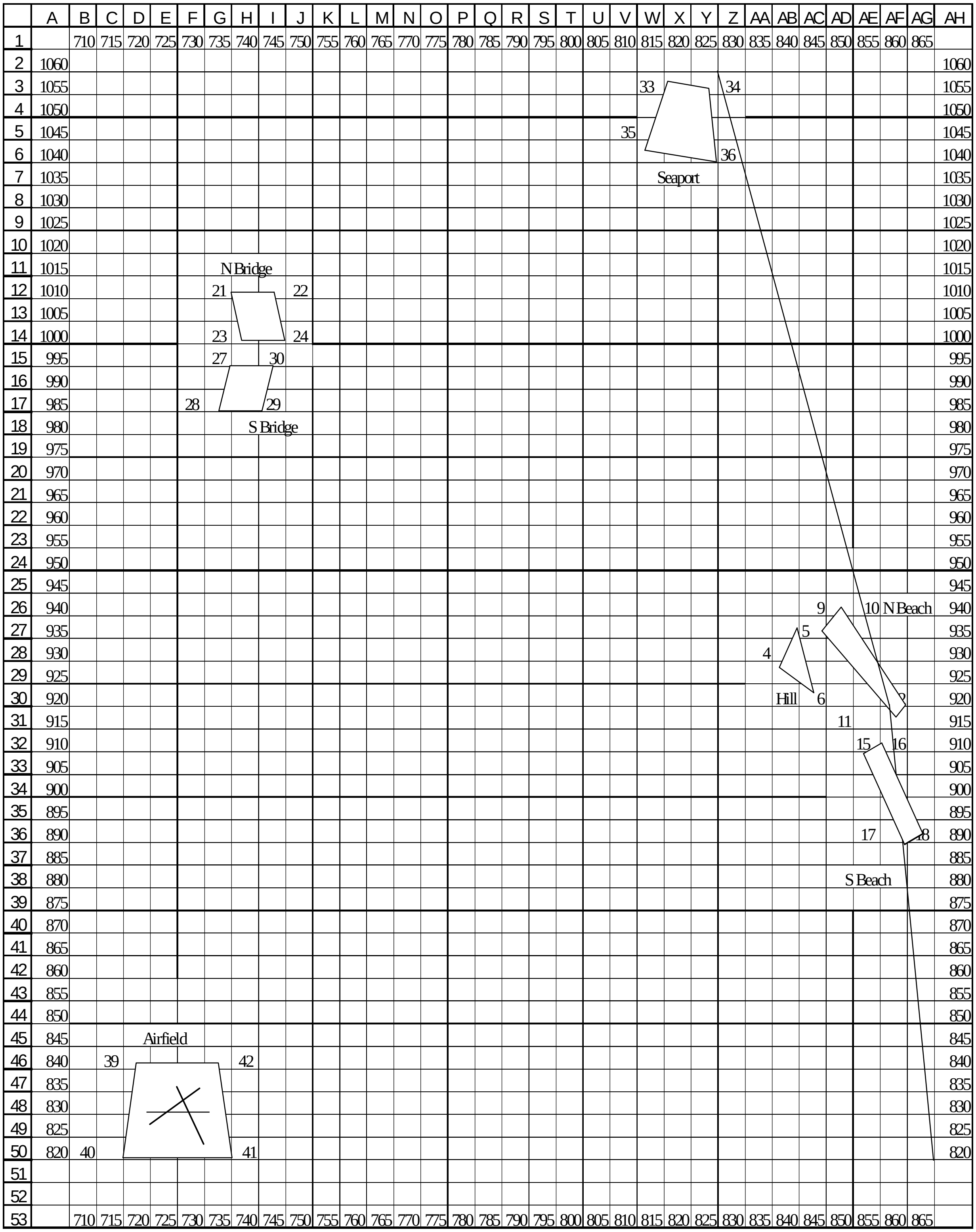

Figure A.8. Map used in data analysis

A.6 **Coordination**

Run #	Date	Decision Maker	Operator	Variant	# of Nodes	Accuracy Score Avg	Coord Score Avg
1	2/23/99	J41	J43	V1	4	88.15%	1.8
2	2/24/99	J43	J42	V1	4	70.30%	1.4
3	2/24/99	J42	J41	V2	4	88.15%	1.4
4	2/25/99	S41	J42	V2	4	93.75%	1.2
5	2/25/99	S42	J43	V2	4	93.75%	1.6
6	2/26/99	J61	J62	V3	6	79.96%	2
7	2/26/99	J62	J61	V4	6	81.22%	1.4
8	3/1/99	J62	J61	V1	6	86.82%	2
9	3/1/99	J61	J62	V2	6	86.82%	2
10	3/2/99	S61	J62	V2	6	90.63%	2
11	3/2/99	S61	J61	V3	6	84.34%	1.8
12	3/3/99	J42	J42	V3	4	78.58%	1.4
13	3/3/99	J41	J41	V3	4	88.15%	2
14	3/3/99	J43	J43	V3	4	93.75%	1.4

Figure A.9. Coordination Score

The coordination was also manually counted and then scored. Number of nodes participating in an event divided by the total possible number of nodes gave the overall team coordination score. The P value was 0.028 and was significant. Coordination requirements were higher for the traditional architecture. See figure A.10 for a Minitab boxplot of coordination scores. The coordination proved to be significantly higher in the six-node architecture.

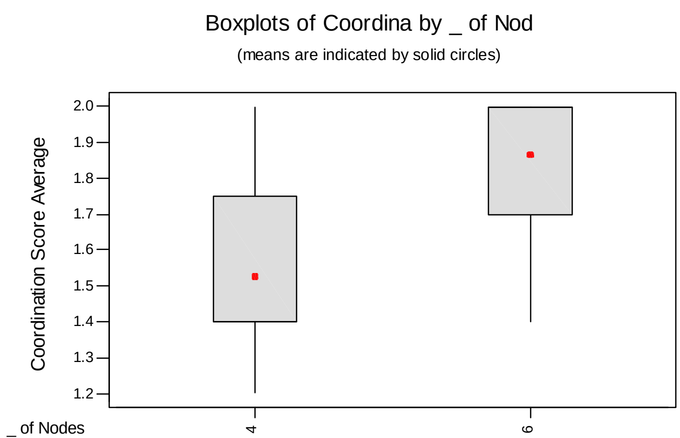

Figure A.10. Minitab Boxplot of coordination scores in experiment five.

A.7 **Sample Red Script**

Figure A.11 shows a sample Red Force script that the OPFOR position would follow; there were four separate variants for the controllers to follow.

Option 1 for controller submitted red forces

D=Dummy H=Hostile N = North Road S = South Road

Silkworm Sites	Lead Vehicle
Silk1 = D	Ld 1 D @ T+3 N
Silk2 = H	Ld 2 D @ T+6 S
Silk3 = H	Ld 3 D @ T+17 N
Silk4 = D	Ld 4 H @ T+20 S
Silk5 = H	Ld 5 D @ T+20 N

A. Medical situations are spawned by the following events

1. Hill
2. Tank2
3. Minefield at seaport
4. Airport

B. Tanks are spawned by the infantry position on the roads.

C. SAM sites are spawned by the infantry units near the airport and the seaport.

Figure A.11. Sample Red Script