NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

MBA PROFESSIONAL REPORT

Analysis of DDD and VDT Simulation Techniques to Determine Feasibility of Using VDT Simulation to Validate DDD Models

By: Eugen Constantin
Nikolaos Papapanagiotou
Sanjeev Singh

June 2004

Advisors: Susan P. Hocevar
David Kleinman
Mark Nissen

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2004	**3. REPORT TYPE AND DATES COVERED** MBA Professional Report	
4. TITLE AND SUBTITLE: Analysis of DDD and VDT Simulation Techniques to Determine Feasibility of Using VDT Simulation to Validate DDD Models			**5. FUNDING NUMBERS**
6. AUTHOR(S) Eugen Constantin, Nikolaos Papapanagiotou, Sanjeev Singh			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this report are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)* The purpose of this MBA project was to determine whether and how VDT can emulate the results obtained from A2C2 Experiments. To do that, we have first focused on learning the basics of VDT and DDD simulation techniques and then on how the models used in DDD can be analyzed using VDT. To this end, we obtained experimental data from DDD Experiment 8 and created representative models in VDT to determine the similarities and differences. We also kept detailed records of our research to assist individuals in the future who may want to expand on our work.

The project involved studying of DDD and VDT techniques, establishing building blocks in VDT, creating a best effort model for DDD Experiment 8 and studying the various outcomes. In this project we could not successfully replicate the complex DDD Experiment 8 scenarios within VDT. However, important conclusions were drawn that would go a long way towards helping future studies in this regard.

14. SUBJECT TERMS Modeling DDD scenarios in VDT			**15. NUMBER OF PAGES** 107
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

ANALYSIS OF DDD AND VDT SIMULATION TECHNIQUES TO DETERMINE FEASIBILITY OF USING VDT SIMULATION TO VALIDATE DDD MODELS

Eugen Constantin, Major, Romanian Ministry of Defense
Nikolaos Papapanagiotou, Major, Hellenic Air Force
Sanjeev Singh, Lieutenant, Indian Navy

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF BUSINESS ADMINISTRATION

from the

NAVAL POSTGRADUATE SCHOOL
June 2004

Authors: ____________________________________
Eugen Constantin

____
Nikolaos Papapanagiotou

____
Sanjeev Singh

Approved by: ____________________________________
Susan Hocevar, Lead Advisor

____
David Kleinman, Support Advisor

____
Mark Nissen, Support Advisor

____
Douglas A. Brook, Dean
Graduate School of Business and Public Policy

THIS PAGE INTENTIONALLY LEFT BLANK

ANALYSIS OF DDD AND VDT SIMULATION TECHNIQUES TO DETERMINE FEASIBILITY OF USING VDT SIMULATION TO VALIDATE DDD MODELS

ABSTRACT

The purpose of this MBA project was to determine whether and how VDT can emulate the results obtained from A2C2 Experiments. To do that, we have first focused on learning the basics of VDT and DDD simulation techniques and then on how the models used in DDD can be analyzed using VDT. To this end, we obtained experimental data from DDD Experiment 8 and created representative models in VDT to determine the similarities and differences. We also kept detailed records of our research to assist individuals in the future who may want to expand on our work.

The project involved studying of DDD and VDT techniques, establishing building blocks in VDT, creating a best effort model for DDD Experiment 8 and studying the various outcomes. In this project we could not successfully replicate the complex DDD Experiment 8 scenarios within VDT. However, important conclusions were drawn that would go a long way towards helping future studies in this regard.

.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENT

We have been very fortunate to have done this work at an institution full of incredibly intelligent and open-minded people! Cross-disciplinary work requires the broadening of horizons, and many people have accompanied us on this journey. We have nothing but admiration for our project advisors: Susan Hocevar, David Kleinman and Mark Nissen, who have been tirelessly enthusiastic and creatively supportive. They each brought their deep expertise to this work, and it is the richer for it.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Adaptive Architectures for Command and Control (A2C2) is a research program sponsored by the Office of Naval Research, made up of a number of teams from firms, defense laboratories and universities including Naval Postgraduate School. They have been collaborating to address issues concerning innovative and adaptive organizations. The main tool used in A2C2 Experiment is known as the Distributed Dynamic Decision-making (DDD) simulation system. This is a distributed multi-person wargaming simulator used for understanding how high-performance teams operate in complex environments. Different from other wargaming simulators, the DDD has a unique flexibility being able to capture the essential elements of different team tasks, to allow the experimenter to vary team structure, access to information, and control of resources. The analytical models and the experimentation tools – most notably the Distributed Dynamic Decision-making (DDD) simulator (Kleinman, 1996) – have undergone considerable refinement, improvement, and extension in order to deal with the increasing complexity of relevant C2 issues. More details about the DDD follow in chapter two.

Results from C2 experiments have shown that "*the better an organization is matched structurally to the overall mission, the better will that organization perform – and that mismatches are potential drivers for adaptation of organization structure"* This is described in organization theory as the concept of "organizational congruence".

The most recent A2C2 Experiment (8), aimed especially to establish the experimental conditions in which the relationship between congruence and performance could be tested (Kleinman et al., 2003).

The basic approach of A2C2 Experiment 8 was to define two disparate organizational structures and then design two missions (scenarios) that exploited the differences between two structures. Then, the next step (Diedrich et al., 2003) was to contrast performance under conditions in which organizational structures and missions were congruent with performance under conditions in which they were incongruent.

The results of Experiment 8 (Kleinman et al., 2003), clearly show that performance in the congruent cases (fit between organization and mission) was significantly higher when compared with incongruent cases (misfit between organization and mission). Performance in this experiment was defined broadly in terms of the number and percentage of tasks completed. Various other measures of performance have been recorded to develop a better understanding of the subject.

In parallel, but without a direct link to the A2C2 research stream, the Virtual Design Team (VDT) research was initiated in the late 1980s at Stanford University, with the goal of *"developing new micro-organization theory and embedding it in software tools that could be used to design organizations"*. The VDT research (Fridsma, 2003; Salazar-Kish, 2001; Fridsma, 2000; Lambert, 2000; Mahalingam, 2000; Kunz et al, 1998; Thomsen, 1998; Jin and Levitt, 1996; Levitt et al, 1994; Christiansen, 1993; Cohen, 1992) aimed to operationalize and extend Galbraith's information-processing framework to model and simulate project organizations. Over the past ten years, the VDT research group has built upon the original framework and developed confidence in the predictions of their theory and tools, which eventually led to the development of SimVision™ – a commercial adaptation of the VDT model.

The primary challenge of this project was to take the VDT out of its commercial use and explore its potential for modeling organizational parameters in a military context. To date, VDT has been used principally in the domain of project management. Attempting to adapt this research approach and toolset to military command and control represents a challenging problem.

Our primary goal in this project is to determine if and how we could use VDT to emulate the results obtained from A2C2 Experiment 8. Building on previous work of both A2C2 and VDT research teams, we also intend to keep records of the various models and behaviors we develop in order to assist people in the future who may want to expand on our work.

Before beginning our analysis we focused on learning the basics of DDD and VDT simulation techniques and on how the models used in DDD can be analyzed with VDT. To understand how DDD really works, we took part as players/decision makers within the ongoing experiments held in the lab. We played several mission scenarios that were either matched (congruent) or mismatched (incongruent) with two organizational structures (Functional, Divisional). We were surprised to see how flexible and user-friendly was the DDD, and "we felt on our own" how performance is reduced in incongruent scenarios. Concurrently we became familiar with the VDT software through VDT tutorials. The four major phases involved in our project are shown graphically in Figure 1.1.

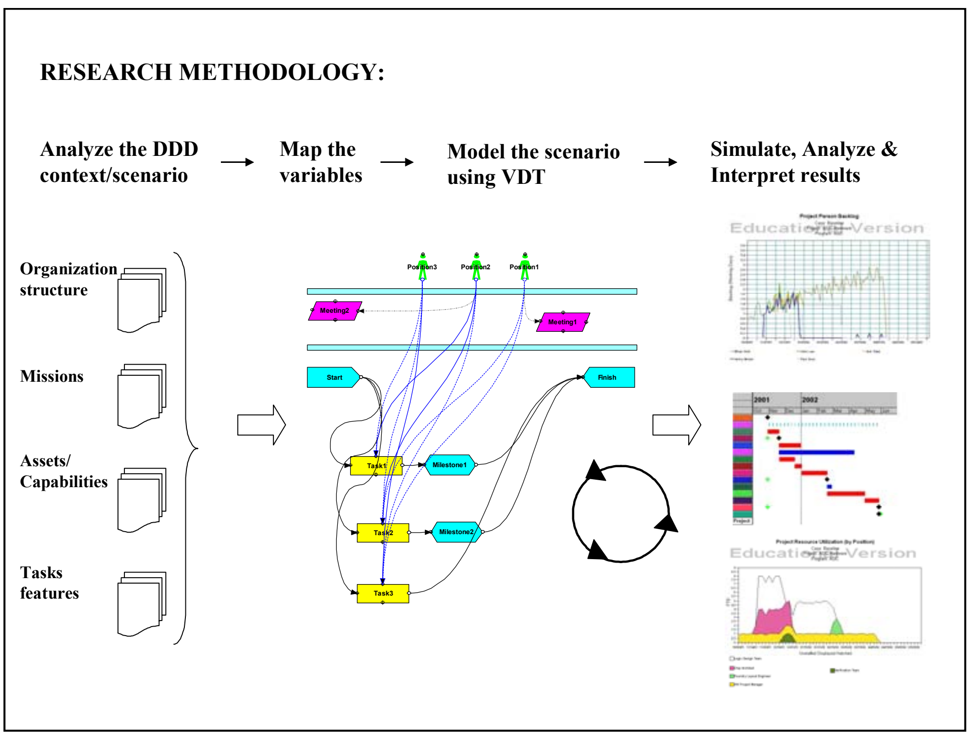


Figure1.1 – The four phases of our team project.

In the first phase, we identified what we considered to be major contexts affecting the congruence/incongruence between organizational structure and mission, using the scenarios and the concluding results from the most recent A2C2 experiment (Experiment 8).

In the second phase, we attempted to map the relevant variables finding the "appropriate match" between DDD and VDT parameters (organization structure, missions, assets/capabilities, tasks etc.). We identified simple but relevant DDD "behaviors" (e.g., resource performing task, contention for resources, delay in resource availability, resource expended, task decomposition, coordination), aiming to replicate them with VDT.

In the third phase we began the modeling process. We started to incorporate DDD contextual behaviors and their local effects into the existing VDT structure. We replicated the identified DDD behaviors and successively refined the VDT models until we got an acceptable match with DDD behaviors. Also in the third phase we started to model with VDT the DDD scenarios: a functional scenario (f) that fits the functional organization (F), while being misfit to the divisional organization (D), and a divisional scenario (d) that fits D but not F. A fuller description of all these efforts is given in Chapters IV and V.

The initial plan for the final phase was to use VDT simulation capabilities for testing and refining VDT models. We also planned to analyze and compare the results against the observed organizational performance during A2C2 Experiments.

This last goal was not reached due to the challenges of modeling critical DDD behaviors with VDT. As a consequence, we limited our work on the final phase to analyzing and interpreting the VDT simulated results and to documenting the gains and limitations of our work, to assist people pursuing this topic in the future and to provide more in-depth examination of how VDT can be used as a pre-experimental modeling tool for A2C2 research. The iterative process established between phase 3 and 4 allow us to analyze results and further refine the VDT models.

In this chapter we have highlighted the starting points, the research goals and the sequence we followed to achieve these goals. The primary goal for this project is to explore the viability of a computational model designed for program management to validate military-specific scenarios and organizations.

In Chapters II and III we review the relevant facts about DDD and VDT, both as a necessary step to Chapter IV and also contributing to a better understanding of the two approaches to modeling and simulation. In Chapter IV we illustrate our efforts to understand the similarities and differences between DDD and VDT. This is a necessary step for incorporating the basic DDD behaviors into the VDT simulation model. In Chapter V we describe in detail the work of building VDT models based on DDD scenarios, including detailed explanations of various models and behaviors. Finally, we conclude by highlighting the contributions of this work, some limitations of the project, and the major conclusions of these modeling experiments.

THIS PAGE INTENTIONALLY LEFT BLANK

II. DDD AND A2C2 EXPERIMENT 8

This chapter presents the salient behaviors of DDD that are relevant to our modeling process. All the material for this section is taken from the following sources: Diedrich et al, “Adaptive Architectures for Command and Control: Toward an Empirical Evaluation of Congruence and Adaptation,” 2002; Kleinman et al, “Scenario Design for the Empirical Testing of Organizational Congruence,” 2003; Diedrich et al, “When Do Organizations Need to Change (Part I) Coping with Incongruence,” 2003 and David L. Kleinman presentation “The DDD. A Team-In-The-Loop Software Tool for Performance Evaluation of Distributed Organizations,” 2001.

A. THE DDD PERSPECTIVE

The DDD is a distributed real-time simulation environment that allows empirical study of team decision making within complex situations. Humans make decisions based on information and resources provided via a simulated framework, and interact in real-time with other distributed team members. The DDD enables the manipulation of variables such as organizational structure and mission scenario within a controlled laboratory environment. The ways in which decision makers (DMs) within a given organization coordinate their information, resources and activities to attain common goals in a dynamic and uncertain task environment has been the focus of A2C2 research for the past ten years.

The first step in the research process abstracts real world problems into a controlled laboratory environment where DMs “play” in the DDD simulation with the team structure (organization) and task environment (scenario) defined by the experiment designer. A variety of performance measures (dependent variables) are recorded such as tasks processed, latencies, and accuracies that allow us to examine the interactions between task or mission structure and the way in which the organization responds to mission requirements. Figure 2.1 displays the elements involved in team decision making.

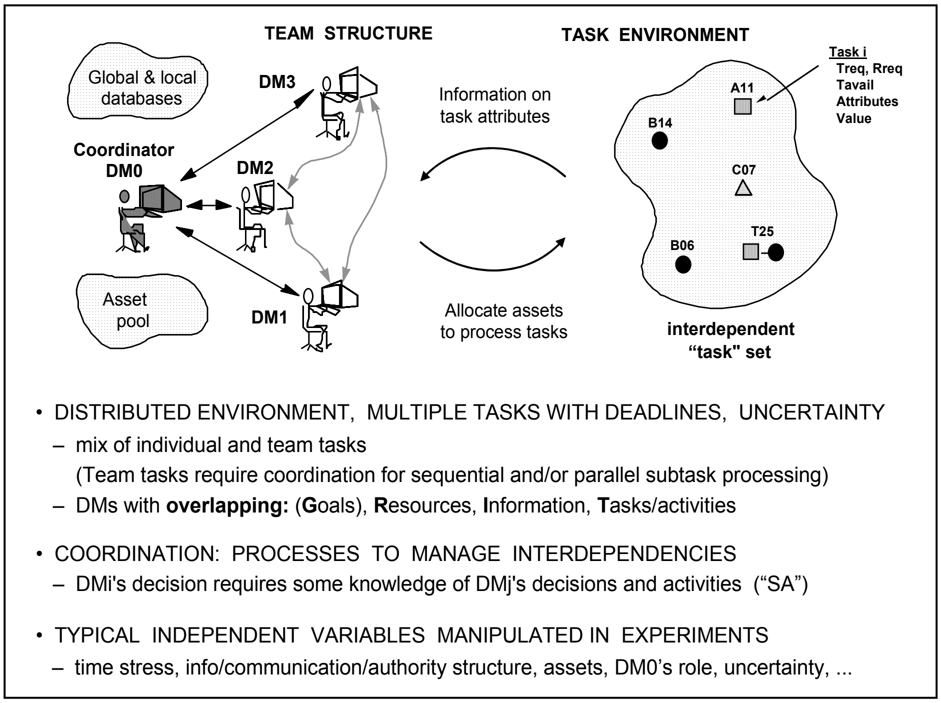


Figure 2.1 – Empirical (laboratory) research in team decision making.

B. ENVIRONMENT

The need to perform efficiently in a Joint battle space led to the design of a DDD paradigm that includes a variety of task classes (e.g., mines, surface threats, strike) and controllable platforms with sub platforms, sensors and weapons (resources) across air, sea and ground. The Joint warfare concept was used to frame the DDD paradigm, with a Joint Task Force (JTF) commander and subordinate DMs, each having responsibility for different but overlapping elements in a common battle space. Under this scenario subjects are required to follow a predefined mission thread ("task graph"), while at the same time defending one or more "penetration zones", and of course their own assets, against potential threats (ground, sea and air). DDD has the ability of to constrain and/or to manipulate organizational structures such as authority, information, communication, resource ownership, task assignment, creates dynamic capabilities for decision making and coordination processes. The overall mission objective must be accomplished through coordination among the DMs in such a manner that allocation of limited organic and nonorganic resources are matched to the requirements of a set of interacting tasks.

C. MISSION STRUCTURE - TASK DIMENSIONS

The scenario designer can create a template of "tasks", with defined attributes, threat capabilities and movements that are linked together by time, space and precedence to provide a theater level mission for the organization to accomplish. An example of a mission structure, or task graph, is shown in Figure 2.2.

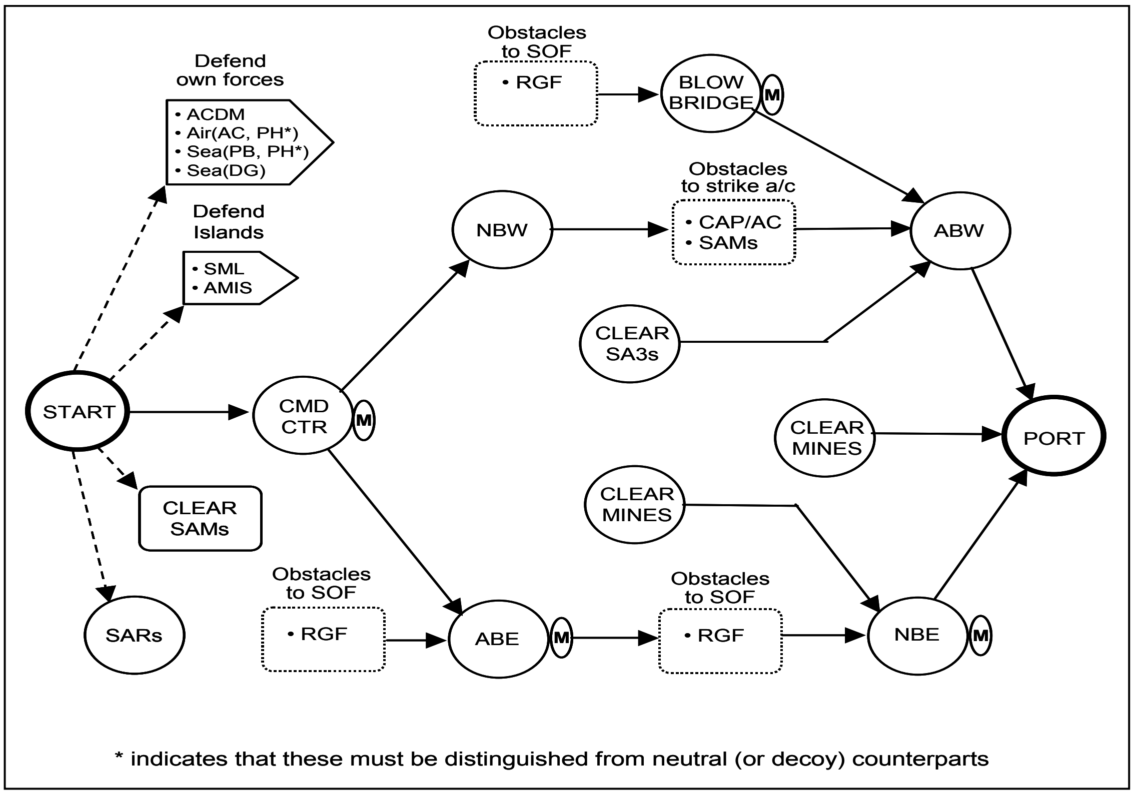


Figure 2.2 – Mission task graph.

Primarily, Figure 2.2 illustrates the task processing sequence. The mission tasks shown as circles are the tasks that must be done and are "known" in advance. This precedence structure indicates the course of action that the players must follow. For example, the enemy command center (CMD) must be destroyed before the capture of the Naval Base East (NBE). In addition, some tasks have prerequisites. For example, mines must be cleared before the NBE and Port tasks can be performed. Also there are ongoing tasks initiated from the start including defend tasks (e.g., self defense) or encounter tasks (e.g., clear SAMs, destroy enemy ground forces).

The characteristics of each task are: type, class, attributes and resources required. Tasks are categorized by type as: air, sea, or ground and are further divided into a variety of classes such as minefields, SAM sites, etc. Attributes are a set of characteristics that define the task and could include speed, hostility, IFF status, etc. Each task has an associated "resource vector" that defines the resources required to successfully process that task. These predefined resources (generic task requirements) are selected by the experiment designer depending on the simulated level of aggregation of the problem.

D. PLATFORMS

Platforms are high value assets that carry sensors and resources (e.g., ships, forward operating bases). The "owner" of a platform controls that platform, and can reposition it geographically and command it to attack. Platforms are also categorized by type and by class.

A platform will often carry sub platforms (e.g., a carrier can contain helicopters and aircrafts) and weapons systems. A sub platform is available for use for a limited time period and could be categorized as either returnable (helicopters), or non-returnable (sonobuoys). Also, sub platforms can be designated as reusable (multiple attacks) or non-reusable (one attack). In both cases (returnable and reusable) there is a time-window that defines a sub platform's availability.

A sub platform (e.g., a fixed-wing aircraft, helicopter, UAV, SOF team) is "launched" from its parent platform, maneuvered to its objective, but has a finite time duration and weapon load (number of "shots") before having to return for refuel or reload. The weapons on a platform (e.g., surface-to-air missiles) are "fire and forget" but are limited in number. The breakdown of resources (platforms, sub platforms and weapons) used in A2C2 Experiment 8 is shown in Figure 2.3.

fixed platforms		mi/min	loadout (subplatforms and weapons)
DDGA	Aegis-capable destroyer	0	6SM2, 2HARP, 8TLAM, 4TTOM, 3ABM, 1FAB, 1HH60, 1UAV
DDGB	Aegis-capable destroyer	0	6SM2, 2HARP, 8TLAM, 4TTOM, 3ABM, 1FAB, 1HH60, 1UAV
DDGC	Aegis-capable destroyer	0	6SM2, 2HARP, 8TLAM, 4TTOM, 3ABM, 1FAB, 1HH60, 1UAV
FFG	Aegis-capable frigate	0	4SM2, 2HARP, 1MH53, 1FAB, 1HH60, 1UAV
CG	Aegis-capable cruiser	0	6SM2, 2HARP, 8TLAM, 3ABM, 1MH53, 1FAB, 1HH60, 1UAV
CVN	Aircraft carrier	0	2F18A, 2F18S, 1MH53, 1FAB, 1HH60, 1UAV
E2C	AWACS - aircraft	0	sensors only - prepositioned for total air surveillance in AOR
FOB	forward operating base	0	3SOF teams preinserted in Country A
AOF	air ops facility on Island E	0	2F18A, 2F18S

subplatforms (reloadable)		mi/min	Tavail	# shots	weapons		mi/sec	range
F18A	air-to-air defender	200	15min	2	SM2	standard surface-air missile	5.0	100mi
F18S	air-to-ground strike aircraft	200	15min	1	ABM	anti-ballistic missile	7.0	85mi
MH53	helicopter mine clearer	40	60min	2	TLAM	Tomahawk cruise missile	2.0	360mi
HH60	search and rescue helo	45	18min	1	TTOM	tactical/steerable Tomahawk	2.0	500mi
UAV	unmanned recon vehicle sensor only	30	60min	n/a	HARP	Harpoon anti-ship missile	1.5	60mi
FAB	fast attack boat	25	20min	2				
SOF	special ops/SEAL team	40	60min	∞				

Figure 2.3 – Platforms, Sub platforms and Weapons

Some of the other information in Figure 2.3 includes the velocity of the sub platforms (mi/min), their endurance or available time, and the numbers of "shots" each can take before needing to return to its parent for reload. The number of shots is equivalent to the number of tasks that a sub platform can process/attack with a full payload. The velocities are approximately ten times real-world values as the game is played at a 10:1 time scale. Some additional information not shown in Figure 2.3 include the sub platform launch and weapon firing delay, and the duty cycle time between successive launches/firings.

Asset (platforms, sub platforms, weapons) capabilities are defined by the resource vector which has the same elements as those associated with the task processing requirements. The total capabilities of an asset package (one or more assets) that is assigned to attack a task must be equal to or exceed the task requirements in order for the attack to be characterized as successful, otherwise the attack will achieve only partial success. A combination of the synchronicity and correctness of the allocated assets is a measure of the effectiveness of the attack.

E. MEASURES AND DATA COLLECTION

A variety of dependent variables are collected by the DDD software that are amenable to study and evaluate different task and organizational structures. Performance measures and process measures are the two basic categories, not necessarily independent, that can be used to tabulate the performance during the experiment.

Performance measures deal with team score and mission effectiveness. These include: team timeliness measures such as latency and slack time; team performance quality includes measures such as accuracy and efficiency in either information processing or resource allocation. In the same manner process measures describe the mechanisms by which the team attained its performance. These measures include the distribution of load between DMs, resource utilizations, as well as communication patterns among DMs.

F. DDD AND A2C2

In recent experiments the DDD was used as an empirical tool to examine the concept of organizational "congruence." The approach for testing the congruence theories was to define two disparate organizational structures and design two missions (scenarios) that exploit the differences in these two structures.

The first mission-scenario was "tuned" (high degree of congruence) to organization 1 and was "mismatched" (i.e., low congruence) with organization 2. The opposite was implemented for the second scenario. The two organizational structures selected for experimentation are commonly referred to as Functional (F) and Divisional (D). In the Functional organization structure, each participant is specialized in one aspect of the mission (such as Strike) using assets that are distributed across multiple platforms (ships). In the Divisional structure, each participant has control over a multifunctional platform and can perform a variety of tasks in a localized geographical region. Thus, a particular mission scenario favored either a divisional or a functional architecture, and was mismatched to the other architecture.

A2C2 experiment number eight (E8) was conducted at NPS in August and November 2002. Participants engaged in a simulation of a mission involving six geographically dispersed commanders in control of assets including six major platforms: three destroyers (DDGA, DDGB, and DDGC), a frigate (FFG), cruiser (CG) and aircraft carrier (CVN) and the related sub platforms and weapons.

The military context for E8 was that country A has invaded and occupied friendly country B and has seized country B's major port (PORT). If attacked, country A has threatened to use tactical ballistic (SCUD) missiles against US allies, island countries D and E. She has also threatened to mine the sea-lanes in order to shut down merchant traffic. Country C could align with country A in opposing US military actions. Enemy forces are concentrated around A's naval bases (NBE, NBW) and air bases (ABE, ABW), and are protecting a major bridge (BR). The enemy has an integrated air defense system (aircraft and surface-to-air missiles), and has surface capability (fast patrol boats and missile-firing destroyers). In addition they have placed their coastal defense missiles in positions that give them maximum standoff against U.S. Navy ships. The military context is illustrated in Figure 2.4.

Joint Task Force objectives are to establish air and sea dominance in the Area of Responsibility (AOR), to prepare the battle space for the introduction of follow-on ground forces, to protect the allies in the region from SCUD missiles, and to protect itself (and neutral shipping) from enemy air and sea attack.

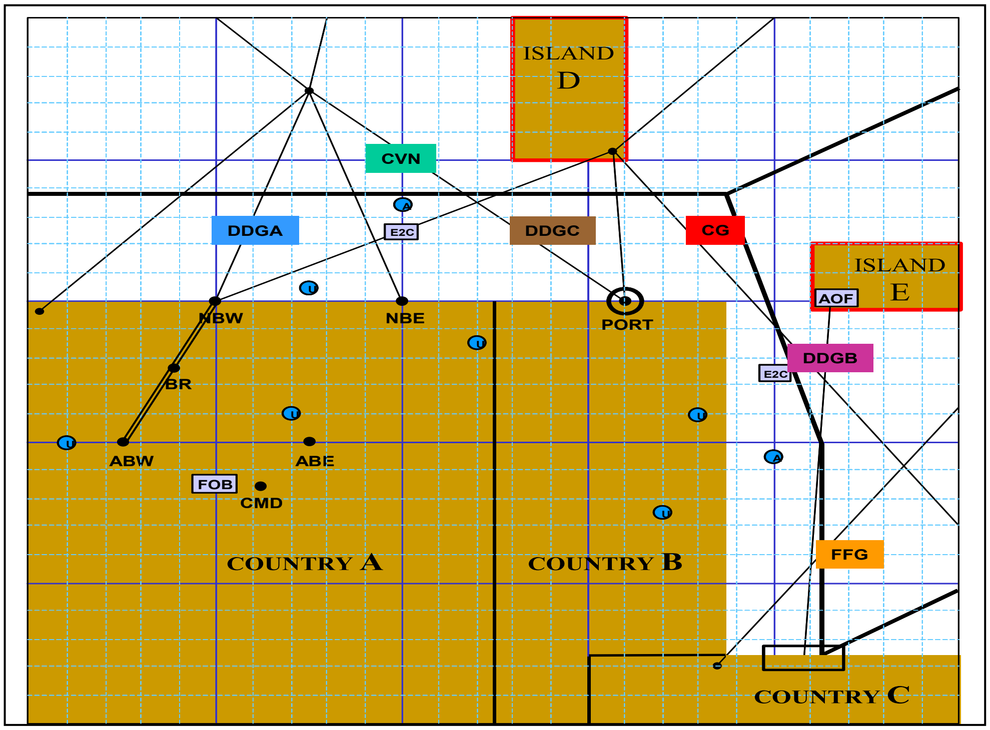
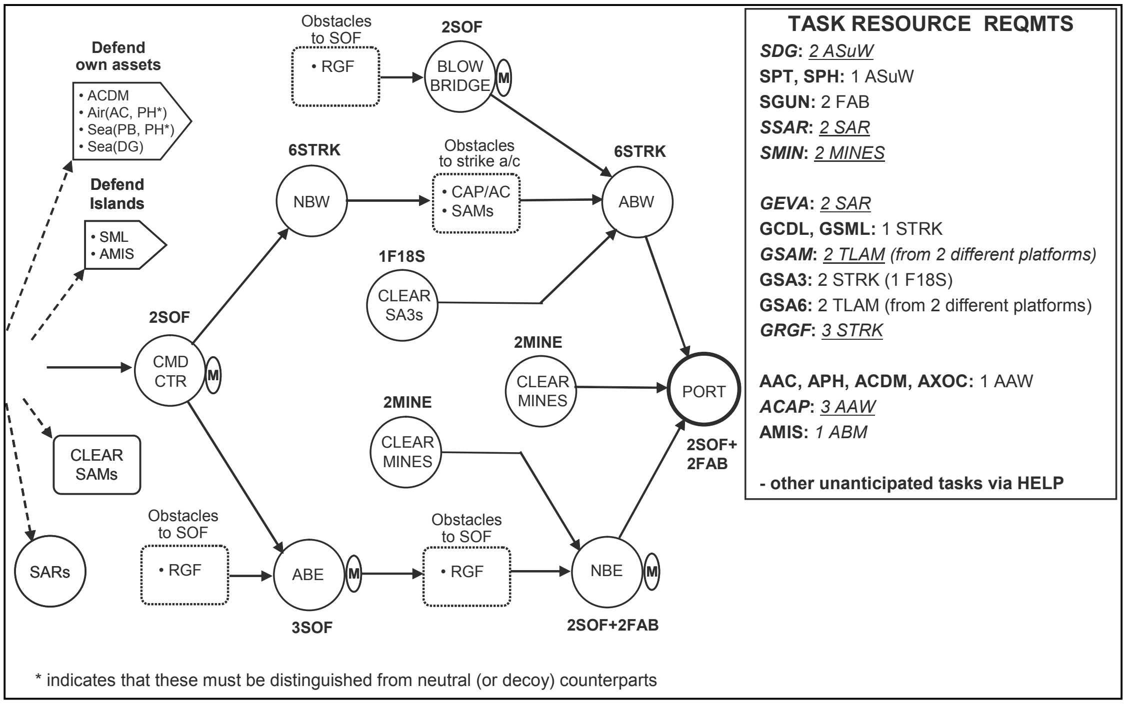


Figure 2.4 – Experiment 8 scenario. Area of Responsibility (AOR).

Figure 2.4 shows the fixed location of the six major JTF assets plus an Air Operations Facility (AOF) on Island E and a Forward Operating Base (FOB) in Country A. Both scenarios have been based on the same task graph, with the salient differences being only in the individual task requirements and enemy force dispositions. To test the hypothesis that congruence between organization and mission leads to good performance, two scenarios were constructed, as shown in Figure 2.5. The first scenario (f) is congruent with organization F and incongruent to organization D while the second scenario (d) is congruent with organization D but misfit to organization F. Of note in Figure 2.5 is that while the tasks (indicated by circles) are the same in both scenarios, the asset requirements for accomplishing the tasks vary. It is this variation that defines the fit or misfit of the two structures with the two scenarios.

TASK GRAPH - A2C2 EXPERIMENT 8 - Scenario f



TASK GRAPH - A2C2 EXPERIMENT 8 - Scenario d

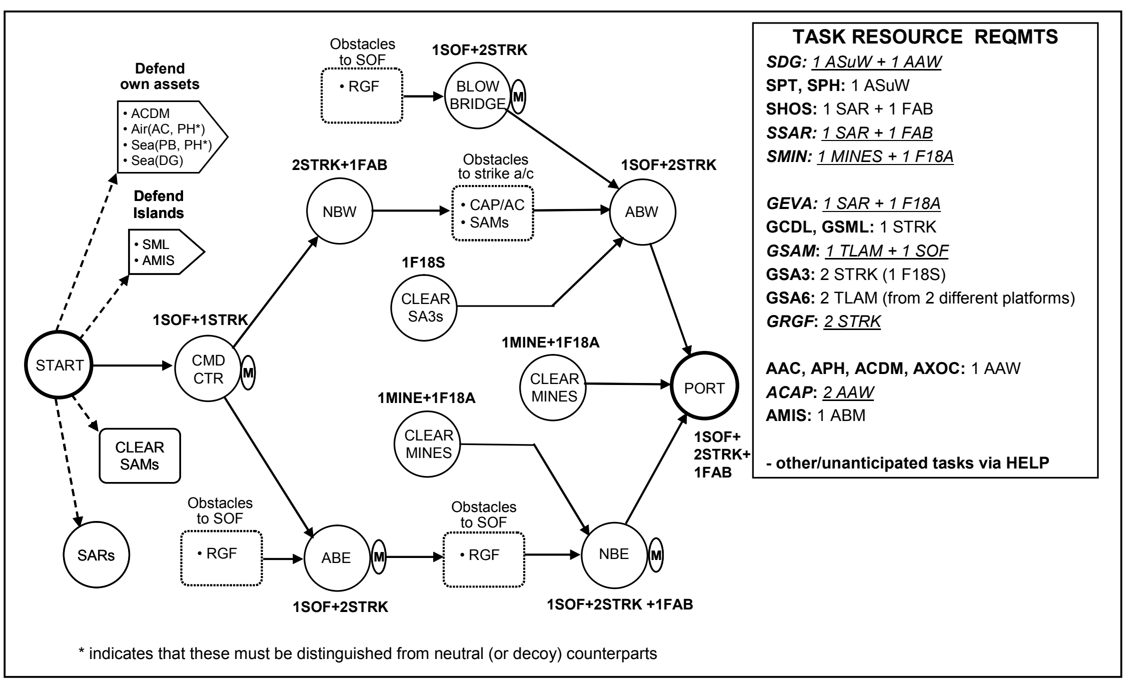


Figure 2.5 – Task Graphs for divisional (d) and functional (f) scenarios.

Both the organizational structures, functional (F) and divisional (D), consists of a six-node matrix describing six specialized missions (strike or surface warfare) and six multiple platforms (ships). As shown in Figure 2.6, the functional structure (F) is organized such that each player (DM) specialized in a single aspect of the mission over the entire battle space, controlling relevant assets that were distributed across multiple platforms (ships). This is reflected by the columns in Figure 2.6. The D organization is represented by the rows in Figure 2.6 and shows that each of the six players owns one of the major platforms with multifunctional capabilities.

Functional

Divisional

DM		1	2	3	4	5	6
	Platform	STRIKE	BMD	ISR	AWC	SuWC/MINES	SOF/SAR
1	CVN	2F18S	xxx	1UAV	2F18A, E2C	1FAB, 1MH53	1HH60
2	DDGA	8TLAM	3ABM,4TTOM	1UAV	6SM2	1FAB, 2HARP	1HH60,1SOF
3	DDGB	8TLAM	3ABM,4TTOM	1UAV	6SM2	1FAB, 2HARP	1HH60,1SOF
4	CG	8TLAM	3ABM	1UAV	6SM2	1FAB,2HARP,1MH53	1HH60
5	FFG*	2F18S	xxx	1UAV	2F18A,E2C,4SM2	1FAB,2HARP,1MH53	1HH60
6	DDGC	8TLAM	3ABM,4TTOM	1UAV	6SM2	1FAB, 2HARP	1HH60,1SOF

* FFGs fixed wing aircraft are located on an island Air Operation Facility (AOF)
SOFs are pre-inserted and located on a Forward Operating Base (FOB)

AAW = anti-air warfare
Mines = mine-clearing operations
ASuW = anti-surface warfare
BMD = ballistic missile defense
Strike = strike warfare
SAR = search and rescue
SOF = special/ground operations
ISR = intel/survellance/recon

Figure 2.6 – Divisional (D) and Functional (F) Organizational Structure. The (D) organizational nodes are depicted horizontally and the (F) nodes vertically.

In the D organization each player is a "platform commander" while in the F organization he/she is a "warfare area commander".

As has been noted, the distinction between the two scenarios (f) and (d) is based upon the resources requirements for the tasks. Therefore, by adjusting the resource requirements of selected task classes, it becomes possible to manipulate what the congruence model postulates, that inter-DM coordination is a major contributor to workload and a detriment to efficiency. Thus, the degree of predicted (structural) congruence is inversely related to the amount of inter-DM (inter-nodal) coordination needed to accomplish the mission. The design of the task classes for both scenarios, as shown in Figure 2.7, made it possible to manipulate the inter-DM coordination needed to successfully accomplish these tasks for the organization-scenario pairing. For example, the resource requirements for the major mission tasks in (f) were adjusted to require coordination among two or more DMs in D but not in F.

symbol	description	AAW	Mines	ASuW	BMD	Strike	SAR	SOF
NBE	Naval base - East	0	0	2	0	0	0	2
NBW	Naval base - West	0	0	0	0	6	0	0
CMD	Enemy command center	0	0	0	0	0	0	2
DG	Missile-firing destroyer	0	0	2	0	0	0	0
PT	fast patrol/missile craft	0	0	1	0	0	0	0
CDL	coastal defense launcher	0	0	0	0	1	0	0
SML	SCUD msl launcher	0	0	0	0	1	0	0
AC	aircraft attack wave	1	0	0	0	0	0	0
ABE	Air base - East	0	0	0	0	0	0	3
ABW	Air base - West	0	0	0	0	6	0	0
SAM	SAM site - fixed	0	0	0	0	2	0	0
ANU	commercial air	0	0	0	0	0	0	0
SNU	white/merchant ship	0	0	0	0	0	0	0
CDM	CD cruise missile	1	0	0	0	0	0	0
MIS	SCUD-launched missile	0	0	0	1	0	0	0
MIN	sea mines	0	2	0	0	0	0	0
XOC	exocet fired at blue ships	1	0	0	0	0	0	0
APH	*possible* hostile air: Yes	1	0	0	0	0	0	0
APH	*possible* hostile air: No	0	0	0	0	0	0	0
SPH	*possible* hostile ship: Yes	0	0	1	0	0	0	0
SPH	*possible* hostile ship: No	0	0	0	0	0	0	0
SA3	*mobile* SAM site	0	0	0	0	2	0	0
EW	possible SCUD launch	0	0	0	0	0	0	0
S&R	basic rescue effort at sea	0	0	0	0	0	2	0
RGF	red ground force	0	0	0	0	3	0	0
SML	SCUD 2nd msl launcher	0	0	0	0	1	0	0
BR	major bridge	0	0	0	0	0	0	2
PRT	final goal - secure Port	0	0	2	0	0	0	2
TSK	enemy hidden airbase	0	0	0	0	3	0	0
TSK	enemy shipping blockade	0	0	2	0	0	0	0
TSK	terrorist leader seen	0	0	0	0	0	0	1
EVA	evacuate wounded	0	0	0	0	0	2	0
GUN	gun runners	0	0	2	0	0	0	0
CAP	aircraft attacker/defender	3	0	0	0	0	0	0
SA6	SAM netted cluster	0	0	0	0	2	0	0

scenario (f)

symbol	description	AAW	Mines	ASuW	BMD	Strike	SAR	SOF
NBE	Naval base - East	0	0	1	0	2	0	1
NBW	Naval base - West	0	0	1	0	2	0	0
CMD	Enemy command center	0	0	0	0	1	0	1
DG	Missile-firing destroyer	1	0	1	0	0	0	0
PT	fast patrol/missile craft	0	0	1	0	0	0	0
CDL	coastal defense launcher	0	0	0	0	1	0	0
SML	SCUD msl launcher	0	0	0	0	1	0	0
AC	aircraft attack wave	1	0	0	0	0	0	0
ABE	Air base - East	0	0	0	0	2	0	1
ABW	Air base - West	0	0	0	0	2	0	1
SAM	SAM site - fixed	0	0	0	0	1	0	1
ANU	commercial air	0	0	0	0	0	0	0
SNU	white/merchant ship	0	0	0	0	0	0	0
CDM	CD cruise missile	1	0	0	0	0	0	0
MIS	SCUD-launched missile	0	0	0	1	0	0	0
MIN	sea mines	1	1	0	0	0	0	0
XOC	exocet fired at blue ships	1	0	0	0	0	0	0
APH	*possible* hostile air: Yes	1	0	0	0	0	0	0
APH	*possible* hostile air: No	0	0	0	0	0	0	0
SPH	*possible* hostile ship: Yes	0	0	1	0	0	0	0
SPH	*possible* hostile ship: No	0	0	0	0	0	0	0
SA3	*mobile* SAM site	0	0	0	0	2	0	0
EW	possible SCUD launch	0	0	0	0	0	0	0
S&R	basic rescue effort at sea	0	0	1	0	0	1	0
RGF	red ground force	0	0	0	0	2	0	0
SML	SCUD 2nd msl launcher	0	0	0	0	1	0	0
BR	major bridge	0	0	0	0	2	0	1
PRT	final goal - secure Port	0	0	1	0	2	0	1
TSK	F14 down & under attack	1	0	1	0	0	1	0
TSK	rescue at POW camp	0	0	1	0	2	1	0
TSK	ship hit mine; under attack	1	1	0	0	0	1	0
EVA	evacuate wounded	1	0	0	0	0	1	0
HOS	hostage taker at sea	0	0	1	0	0	1	0
CAP	aircraft attacker/defender	2	0	0	0	0	0	0
SA6	SAM netted cluster	0	0	0	0	2	0	0

scenario (d)

Figure 2.7 – Task resource requirements

The results obtained from E8 demonstrated that performance in the congruent cases significantly exceeded that in the non-congruent cases. Based on the comparison of the matched cases (D-d and F-f) with the mismatched conditions (D-f and F-d) the results show that communications and workload were increased in the mismatched conditions. Finally, participants in both organizations processed fewer tasks in the incongruent cases. Experiment 8 demonstrated the power of model-based organizational design for optimization of mission effectiveness.

THIS PAGE INTENTIONALLY LEFT BLANK

III. VIRTUAL DESIGN TEAM (VDT)

The VDT (Jin and Levitt, 1996; Kunz et al, 1998; Levitt et al, 1994) research was initiated at Stanford University in the late 1980s. The long term goal of the research is to develop computational tools to analyze decision making and communication behavior within organizations.

A. INTRODUCTION

The basic premise of VDT is based on information processing view of organizations (Galbraith, 1977). According to Galbraith (1977:3), organizations are "*composed of people and groups of people in order to achieve some shared purpose through a division of labor integrated by information-based decision processes continuously through time*." Organization design involves bringing these factors together with the choice of strategy, that is, searching for a 'fit' between strategy, method of organization and integration of individuals (Galbraith, 1977:5). Galbraith put forward arguments to the effect that "*the greater the task uncertainty, the greater the amount of information that must be processed among decision makers during task execution in order to achieve a given level of performance*" (Galbraith, 1977: 36).

VDT operationalizes and extends Galbraith's framework to generate specific predictions about information processing capacity versus load at the level of individual actors or subunits. Actors send and receive messages to and from other actors along pre defined channels of communication, which in turn are dependent upon the exceptions generated during processing of tasks. Each actor has a queue of information tasks to be performed (e.g., work activities, attending meetings) and a queue of information outputs (e.g., completed work, request for assistance) (Nissen, 2003). Actors may be constrained by a number of factors like number of individuals in that position, work volume, and downtime. Performance is directly affected by factors such as how well the skill set of the actor matches those required for the task, his/her backlog and the relative priority of the task.

VDT also quantifies the effect of a variety of organizational decision-making policies (e.g., the level of centralization of decision-making) on the handling of exceptions and the likelihood that they will be responded to. VDT thus generates specific predictions about activity and project schedule, cost and work process quality for a given organization assigned to specific project tasks. Relatively simple, high-level VDT models may contain five to ten business milestones, each enabled by a few tasks, assigned to fewer than twenty organizational actors. These relatively sparse models describe complex projects in sufficient detail to make extremely accurate predictions about schedule, cost and quality performance. Figure 3.1 shows the inputs and outputs of a VDT model.

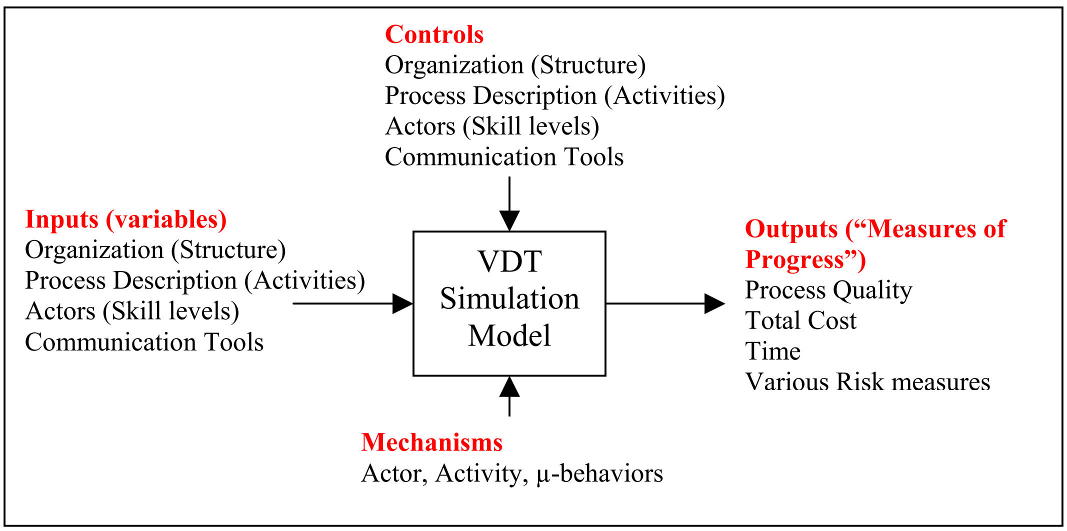


Figure 3.1 – **VDT Model Architecture**. Given values for independent input variables that describe a project and a set of fixed assumptions, the VDT model simulates each activity being performed by responsible actors and computes overall project duration, cost and coordination quality. The micro behaviors consider both planned direct work and inferred requirement for coordination and rework.

For a particular analysis, a set of organizational attributes are held fixed as control variables and a small set of variables are changed as independent variables in the simulation. The fixed attributes are defined in behavior matrices that have been developed based on real world organizational behavior. For example, with low centralization, the decision about whether to perform rework when an error is detected is made by the actors themselves while in the case of higher centralization the decision would be referred to project manager. VDT also allows users some control over these variables. Variables like skill level and degree of centralization can be set by the user. Organization (Structure), Process Description (Activities), Actors (Skill levels), and Communication Tools depend upon fixed as well as user defined attributes. Hence they are listed under Controls as well as Inputs in Figure 3.1. The fixed aspects are represented as Control and the variable aspects are represented as Inputs.

The VDT system is an object oriented, discrete event simulation tool. The model uses inheritance and behavior methods using symbolic pattern matching. The VDT discrete event simulation of stochastic behaviors uses Monte Carlo simulation. It uses editable decision tables (behavior matrices) to transform qualitative attribute values to quantitative values that are used in discrete simulation. The degree of centralization of decision making responsibility affects the outcome of the simulation. It is used to determine who will take decisions about whether to perform rework when an error is generated. The decision table also determines the probabilities that managers at each level decide to rework, correct or ignore the error. The interplay between symbolic inference and numerical Monte Carlo simulation enables VDT to replicate Galbraith's predicted behaviors for an organization (Levitt et al., 1996).

B. OVERVIEW OF MODEL BUILDING

SimVision™ is a commercial implementation of the VDT model by Vité Corporation that is licensed by ePM, Inc. This project uses SimVision™ (henceforth referred to as VDT) to emulate organizational behaviors from DDD Experiment 8.

VDT is a complex modeling environment and it is the aim of this chapter to introduce basic concepts to the reader. To know about VDT in detail the reader is urged to refer the SimVision™ user's guide (Revision 7: January 15, 2003).

VDT models consist of two major components: task structure and organization structure. Task structure defines the work to be done and how individual tasks are related to each other while organization structure defines the shape of the organization (e.g., degree of centralization, teams, and sub teams). VDT provides great flexibility to change various parameters for both these structures in order to customize the model to suit user requirements. The simulated outcome depends on how well the complexities of the tasks are matched to the capabilities of the responsible positions.

VDT employs graphical components (e.g., Positions (human shapes), Meetings (parallelogram), Tasks (rectangles), Milestones and links) to build models (Figure 3.2). Objects used to represent work, events, and personnel in the model — such as projects, tasks, milestones, meetings, organizations, and positions — are referred to as shapes. The various means used to connect shapes to one another in order to express their relationships are called links. Users can define properties for each of these shapes and links.

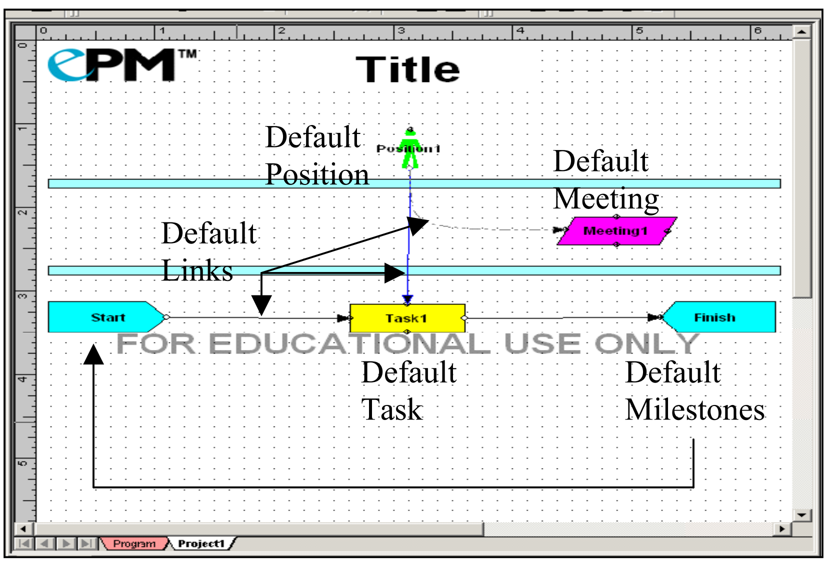


Figure 3.2 - Basic building blocks of a project

A model consists of the model or program page. A program is a set of related projects that together produce the product or products. It also includes the associated responsible organizations, milestones, and relationships between projects. For example, consider development of an IT product that requires a software and a hardware component. Both these components may be represented by separate projects within the same program. Similarly, there can be two organizations representing software and hardware developers separately. Organizations may be linked to specific projects or left independent. If left independent, the persons within organization are available for all the projects within the program whereas if assigned to a project, they are available for work only for that project. Each project and organization has its own window. Milestones are used to define waypoints in the program and can be used to group a set of tasks (Figure 3.3). Two milestones, Start and Finish, are defined by default in each program and project window.

A project is a set of related product development tasks including positions (groups of one or more individuals) that perform tasks and the dependencies between tasks and positions. Each project is represented in its own window (Figure 3.2). Tasks are linked together for either sequential or parallel execution depending upon the objective. They are specified by the volume of work required (defined in man hours, days or weeks) and the skill required for completion of the task. Positions are represented in the projects using graphical human shapes. Each position represents a group of people who may be defined by either the number of FTE's (Full Time Equivalent people) or by staffing individuals from departments. The positions in an organization are linked together in desired hierarchy using Supervision links. Each task is assigned to one or more positions with one of them assuming primary responsibility for the task. Each position, in turn, could be assigned to one or more tasks that are executed depending on the task sequence and priority (Figure 3.3).

Organizations, like projects, are represented within their own window. Each organization can have one or more departments that can be staffed with people. Persons within departments are associated with properties such as name, number of FTE's and one or more skills. They are used to staff positions in the project window.

Positions or persons can be assigned to meet using the Meeting object. Meetings are gatherings of actors to communicate about the project and project tasks. One or more actors can be assigned to meet at predefined times and intervals (Figure 3.2). These meetings depend upon and affect the amount of coordination required. However, they also increase work backlog.

A link shows a relationship between two objects and any dependencies one object has on the other. The various links available include task links, rework links, communication links, supervision links, assignment links and meeting links. Some of the major links are shown in Figure 3.2.They are used to link tasks to tasks, positions to tasks, or positions to meetings. They enable us to create sequential, pooled or reciprocal dependency among tasks.

Models can be simulated once they have been constructed (Figures 3.3 and 3.4). The simulator results include the following information:

- The predicted time to complete a project.
- The total effort required to complete the project.
- Several measures of process quality.

VDT offers extensive text and graphical analysis of the model. Using charts and data that result from simulation, one can identify risks in the project, organization, and performance. Some of the graphical representations include Program and Project Gantt charts, Summary statistics, Schedule growth chart, Breakdown chart, Coordination chart and Quality risk chart. Planned and simulated project progress is presented using Gantt Charts (Figure 3.6). The user can also view position and person backlogs to identify overloaded positions or persons. Model analysis using VDT helps in pre project strategic planning, effective organization design and staffing, periodic program and project checkup and process refinement.

C. HOW TO BUILD AND ANALYZE MODELS

Model building starts with creating a new program in VDT. The default program consists of a default project linked with Start and Finish milestones. The default project includes a default task, a default position, default milestones (Start and Finish), and a default meeting (Figure 3.2). The user can then add additional objects depending upon the requirement.

Once the basic model has been constructed (called the baseline model where all error probabilities [see below] are set to zero), it is checked against predicted outcomes by simulating the model. If the simulated results match the predicted results the baseline model is assumed to be correct. Further enhancements to the baseline model can be done by adding communication and rework links and setting appropriate values for the error probabilities. These links make the model more representative of some real-world applications by introducing elements of communication, interdependency and coordination within the project.

To quantify the distractions in a model such as the exchange of information, telephone calls, impromptu meetings, and rework caused by failed tasks, four different probabilities (Information Exchange Probability, Noise Probability, Functional Error Probability, and Project Error Probability) can be set in the program window. The information exchange probability measures the level of communication in the project between positions that are responsible for tasks linked by communications links. Noise is a way to measure the effect of interruptions in the ordinary working day that take time away from doing the project tasks. Functional error probability is the probability that a task will fail and require rework. Project error probability is the probability that a task will fail and generate rework for all dependent tasks connected to it by rework links.

A sample model developed as part of this project is shown below.

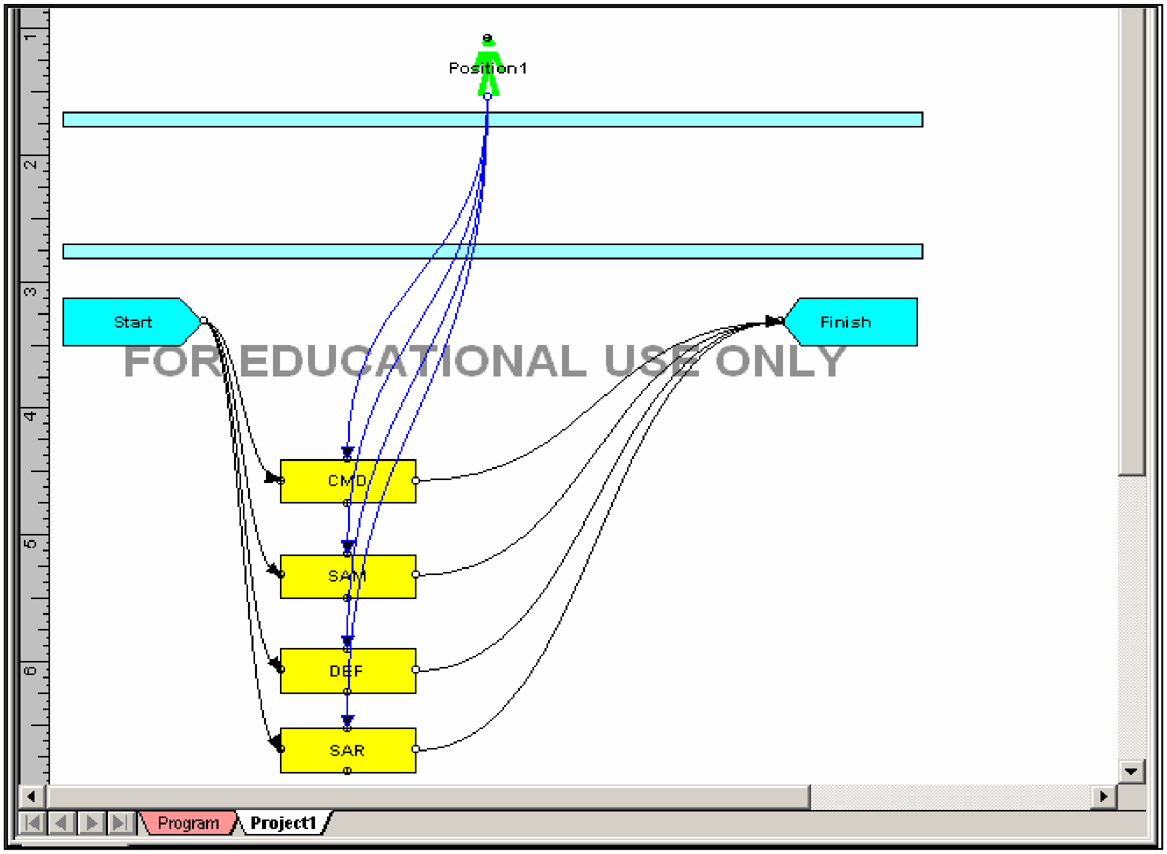


Figure 3.3 - A sample model

The model in Figure 3.3 comprises four parallel (simultaneous) tasks. A position (Position 1) is responsible for executing these tasks. Each of these objects has various properties that can be set in the properties window. The variable properties for a task that can be set are seen in Figure 3.4.

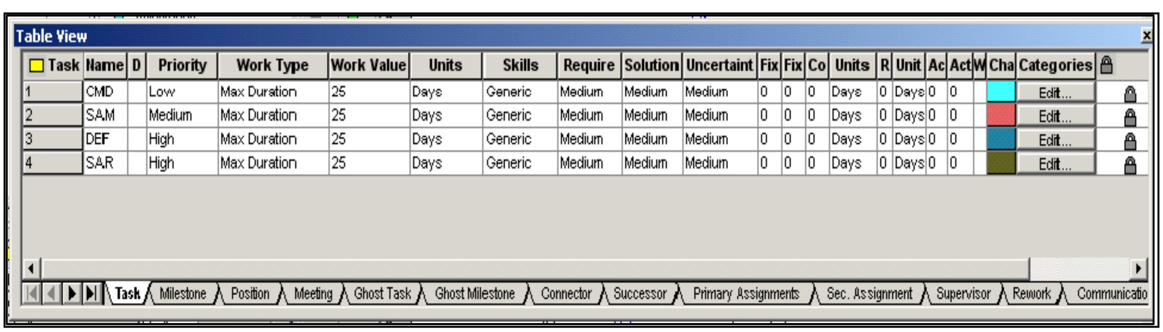


Table View

Task	Name	D	Priority	Work Type	Work Value	Units	Skills	Require	Solution	Uncertaint	Fix	Fix	Co	Units	R	Unit	Ac	Act	W	Cha	Categories	
1	CMD		Low	Max Duration	25	Days	Generic	Medium	Medium	Medium	0	0	0	Days	0	Days	0	0			Edit...	
2	SAM		Medium	Max Duration	25	Days	Generic	Medium	Medium	Medium	0	0	0	Days	0	Days	0	0			Edit...	
3	DEF		High	Max Duration	25	Days	Generic	Medium	Medium	Medium	0	0	0	Days	0	Days	0	0			Edit...	
4	SAR		High	Max Duration	25	Days	Generic	Medium	Medium	Medium	0	0	0	Days	0	Days	0	0			Edit...	

Task | Milestone | Position | Meeting | Ghost Task | Ghost Milestone | Connector | Successor | Primary Assignments | Sec. Assignment | Supervisor | Rework | Communicatio

Figure 3.4 – Task Properties

The various properties associated with tasks include name, description, priority, work type, and skill among others. Priority is the importance of a task relative to others in the project, expressed as High, Medium, or Low. By setting a task's priority, one can influence how a position will treat it compared to other tasks occurring in parallel (i.e., simultaneously). Positions tend to attend to tasks in order of priority. Tasks with highest priority tend to receive attention before lower priority tasks. Thus, highest-priority tasks tend to finish before lower-priority tasks.

A task's work volume determines how long it is expected to take. There are three types of work volume to set for a task: Work Volume, Work Duration, or Max Duration. Work volume is the amount of work that needs to be done to complete the task. If more FTE's are assigned to the task then it will be completed in lesser time. Work duration is the amount of time the task takes. Work duration is used for tasks whose duration is determined by factors other than the number of FTEs assigned. Rework and coordination can cause the task to take longer than specified when specified in terms of work duration. Max duration is used for tasks whose duration is determined by factors other than the number of FTEs assigned. These tasks are not constrained by their work volume but by time, and where coordination and rework do not affect the task.

Skill is the primary area of expertise needed to perform a task. The default skill is Generic, indicating that the task requires only those abilities possessed by the average worker. Other skills can be added to the project, thus enabling selection of a skill more appropriate to the task's requirements. In the above project all the tasks have a differing priorities, generic skills and work type property is set to max duration.

Various properties associated with positions can be seen in Figure 3.5. They include, among others, name, description, role, application experience, FTE, skill set and staffing. The role describes the position's organizational function on the project team. It also affects the types of decisions that are made by that position. There are three types of position roles: Project Manager (PM), Subteam (ST) and Subteam Leader (SL). Usually only one person is appointed PM in a project. The Subteams actually do most of the work. This is the default role. The ST passes some exceptions to the Subteam Leader (SL), which in turn passes some exceptions to the PM. All positions between ST and PM are designated as SL. The PM normally generates more rework than the SL, which in turn generates more rework than the ST. Application Experience describes how experienced the position is with this type of project. There are three values for application experience: high, medium and low. Because of its effect on position work processing speed, application experience has a dramatic effect on project duration, internal and external exceptions. The position FTE value shows the number of full-time equivalent (FTE) people that an unstaffed position represents. A position can represent multiple full-time people, part-time people, or a combination of both. The Skill Set displays the skills that a position has available and the position's skill level (high, medium or low) in each of these skills. Staffing represents the person or persons assigned to the position. These persons usually belong to departments that are defined in the Organization window. In the above project (Figure 3.3), Position 1 is an unstaffed ST, has medium application experience, generic skill, and a FTE of one.

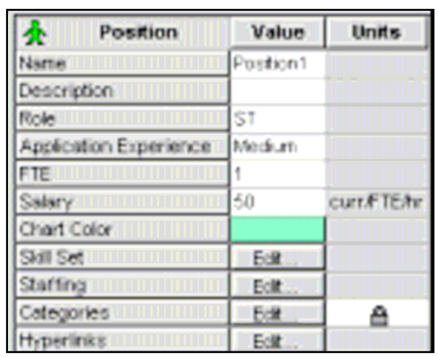


Figure 3.5 – Position properties

Simulated result for the above model is as shown below.

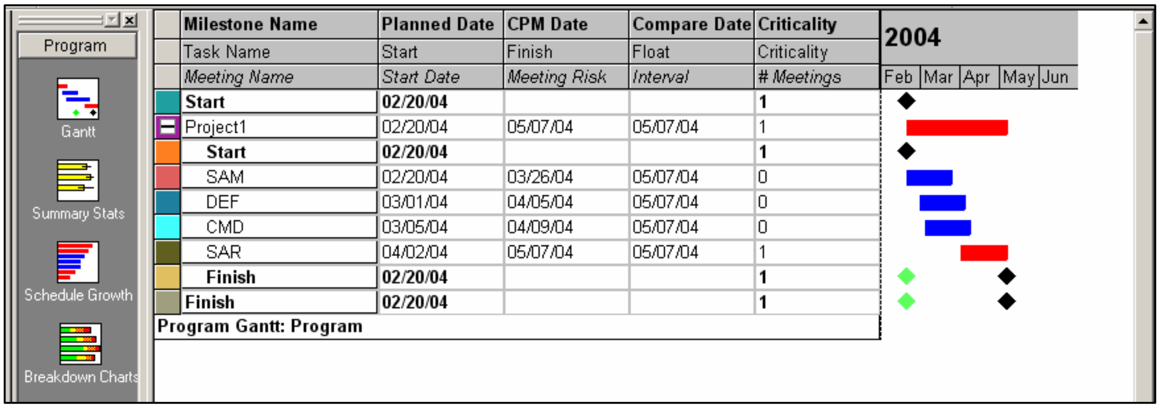


Figure 3.6 - Simulated results for the sample model

The simulation provides comparison between the planned dates, simulated dates (not in figure) and the critical path (CPM) dates. Planned dates are the dates assigned by the user and are run through the simulator only to allow for comparison against the simulated and CPM dates. The principal difference between simulated and CPM results is that simulated results take into account all the real-world factors that have been modeled, whereas CPM results reflect an optimistic scenario where all positions are fully available to work all the time on all their tasks unless the tasks overlap. For example, simulated results factor in time for communications between positions about tasks, exceptions in tasks, and the resulting rework and possible backlog for the responsible positions. CPM results ignore all these factors. VDT also provides various other charts and data to interpret the simulation results. In addition to Gantt charts, the user can check Project summary, Person and Position backlogs, Project breakdown, Project schedule growth and various risks associated with the project.

VDT offers numerous features to model and control various aspects of a project or a program. In our project however, we will restrict ourselves to using some of the basic features and allow the software to keep default values for all others.

D. CONCLUSION

The VDT framework is ideal for this project because it distinguishes individuals and their roles, represents dynamic changes in the organizational work environment, and provides detailed traces of participants executing work process and communication tasks. The primary challenge of this project is to take the VDT out of its commercial use and explore its potential for modeling organizational parameters in a military context.

IV. SIMILARITIES AND DIFFERENCES BETWEEN DDD AND VDT

The previous two chapters have described the main simulation tools that are central to our project: the Distributed Dynamic Decision-making (DDD) simulation system – a real-time, team-in-the-loop war-gamming environment, and SimVision^{TM} – a commercial implementation of the VDT (Virtual Design Team) *simulation* model, used for projects management.

We considered VDT well-suited to study the effects of context on organizational performance as it includes basic elements that represent individual actors, their decision-making preferences, the organizational structure, decision-making policies and performance metrics. In addition, both DDD and VDT are relatively mature products that have demonstrated excellent utility in past applications. Both products appeared to exhibit many synergisms that offered possibilities for exploitation in our research. However, there were also several differences that, if not resolved, could adversely affect our objective of applying VDT to the DDD environment.

During our participation in DDD Experiment 8, we observed six decision makers (DM's) playing a war-gaming scenario in which the DM's communicated over a voice network in order to coordinate team activities. Supposedly VDT could be used to model this scenario, and the team coordination processes, within a hypothetical model.

VDT was designed to simulate alternative options (by planning, running the simulation, acknowledging how feasible the plan was, having the opportunity to go back and refine it, etc.). In contrast, in DDD, instead of simulating multiple alternatives, the alternatives are in the minds of the players and only a *single* alternative can be played in any real-time run. Thus, the A2C2 research team saw potential value in being able to *model* alternative architectures using VDT in order to complement A2C2 research wherein military officers *play* alternative architectures in a DDD simulation.

A. LEARNING THE FUNDAMENTALS: MODELING BASIC TASKS

In an attempt to understand the DDD/VDT differences, we started by comparing the two tools with the aim of finding the appropriate match between their relevant operating parameters. Running SimVision tutorials, we learned that some VDT features exhibited a very straightforward correspondence with DDD. Both DDD and VDT had "Start" and "Finish" points, "Tasks" and "Milestones" in a sequence dictated by the planned/modeled scenarios. This similarity is illustrated in Figure 4.1 where we compare a simplified task graph (abstracted from the scenario used in A2C2 Experiment 8) with a default project sequence, as it appears in the VDT model pane.

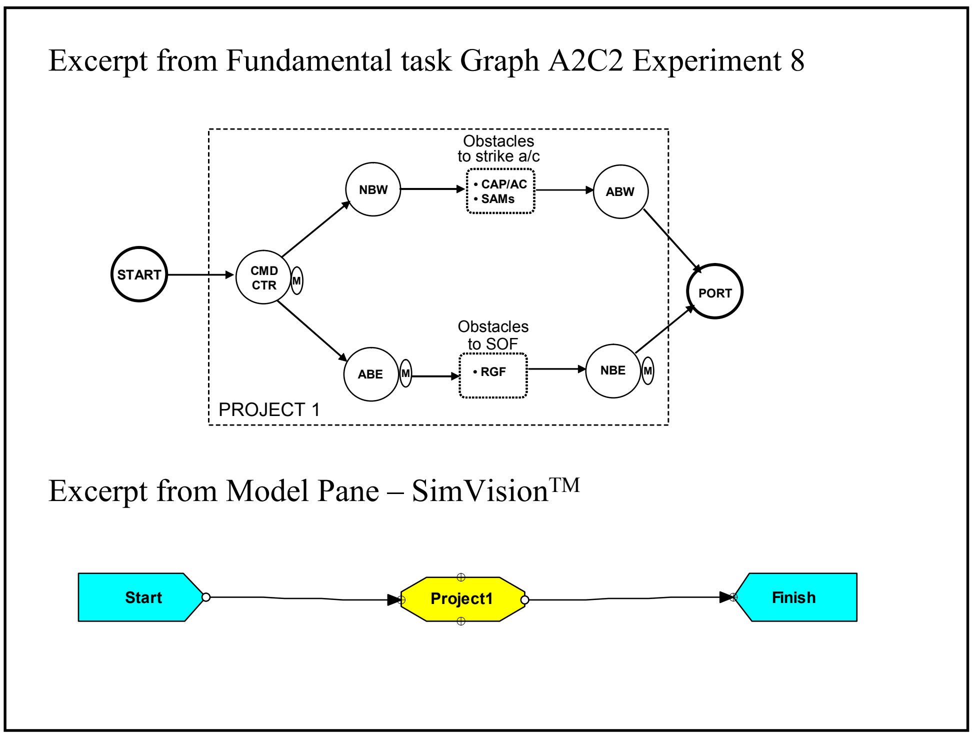


Figure 4.1 – Simple correspondence between a task graph scenario used in A2C2 Experiment 8 and a default project sequence – VDT.

The primary mission tasks in Experiment 8 are shown on the task graph as circles linked with "finish-start" arrows in a planned sequence, having an initial "START" point, and a final objective - "PORT". In VDT, we also have "Start" and "Finish" points, interconnected with "Project 1" box using "finish-start" arrows. The "Project 1" box can be "decomposed" further to model the same sequence of tasks illustrated in the task graph of Figure 4.1. The primary tasks shown in Figure 4.2 by rectangles are interlinked with milestones, using the same "finish-start" arrows.

Similarly, a number of different kinds of tasks in the Fundamental Task Graph of Experiment 8, such as: *primary tasks* (the enemy command center – CMD CTR, the enemy air bases – ABE/ABW, the enemy naval bases – NBE/NBW, and the final objective-PORT), *time-critical tasks* (destroying SCUD launchers, search and rescues, etc.), *offensive prerequisite* (e.g., mine clearing), or *defensive tasks* - could be easily replicated within VDT.

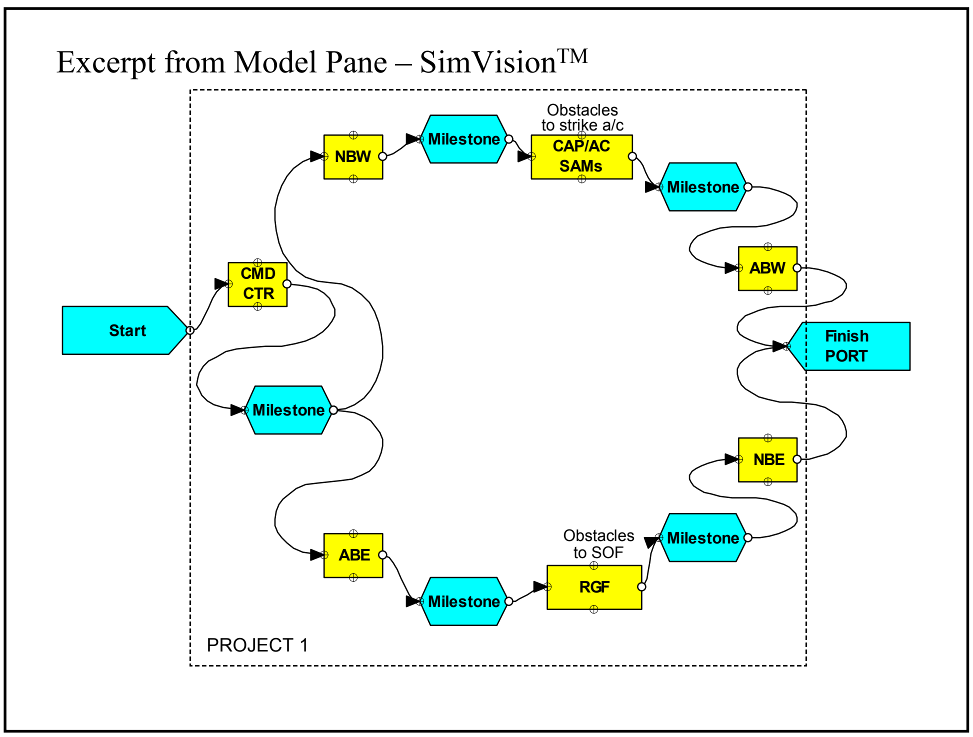


Figure 4.2 – Detailed VDT "Project 1" box, following the same sequence as in the simplified task graph – A2C2 Experiment 8, depicted in Figure 4.1.

As briefly described in the previous chapter, task precedence can be very simply modeled in VDT using successor links: "finish-start" (the successor task starts when the predecessor task finishes, or also a specified time delay following finish), or "start-start" links (the successor task starts at the same time as, or a specified time after the predecessor task starts).

At this point, after envisioning success in modeling the mission tasks, we also thought that time-critical tasks could be placed in the sequence as planned in DDD, and that VDT should be able to handle the assets for these tasks. In general, for some of the VDT parameters we could see some fairly clear links to parameters in DDD, for others these links were less obvious. In some cases, however, the two differed more substantially. For Example, in DDD/Experiment 8 the decision makers were not hierarchically differentiated as they are assumed to be in VDT. Experiment 8 used a "flat" structure, not an organizational hierarchy. Furthermore, the positions' training/experience, which intuitively affects the results, is not included within the DDD model but is part of VDT.

B. MODELING THE ASSIGNMENT OF ASSETS/RESOURCES TO TASKS

Another representational issue was that DDD/Experiment 8 assigned assets to tasks rather than positions to tasks, as done in VDT. The basic elements of the VDT model include the decision makers/positions in an organization, and the tasks that comprise a work process. Each position has attributes such as skill set, level of experience, and organizational role.

Our initial view was that we could define parameters in VDT such that positions would be characterized as assets/platforms (e.g., Aircraft carriers – CVN; Aegis-capable destroyers – DDG), "staffed" with different sub platforms or weapons (standard surface-air missiles – SM2; anti-ballistic missiles – ABM), and having different "skills" (SM2 – "anti-air" skill etc.). Furthermore we assumed that we could assign assets to primary and secondary tasks using the primary and secondary assignment links (as described in Chapter III). We also planned to use the VDT "FTE" feature (the number of full-time equivalent persons represented by the position) to establish the number of sub platforms/weapons under a certain asset/platform.

An additional concern was related to "task work volume", which in VDT represents the amount of work one predicts the task will require. There are various ways to express this "work volume" – person hours, weeks, and months. However, this work volume represents only the *direct* work of the task; during VDT simulation, the task may require more time due to coordination tasks and rework. These comprise the information-processing requirements of the task. The closest parameter to work volume in DDD is "workload demand", used by the modelers to balance asset management and effort among DM's. Each DM is prompted by the DDD simulator to provide an "estimate of the workload they experiencing" (Diedrich et al., 2003) as a way of data gathering on workload assessment during scenario. Based on the modeling approach and the manipulations relative to the DDD congruent and incongruent cases, the overall level of perceived workload should be higher in mismatched cases.

Another resource parameter that we attempt to find a DDD correspondence with is referred to in VDT as "skill", that is the ability of a person to perform a specific task. In DDD/Experiment 8, the six primary platforms did not have any native or organic capabilities – all of their organic capabilities derive from their sub platforms and weapons systems. We could model this in VDT, by assigning a "generic" (general) skill to the primary platform while assigning specific skills to the organic sub platforms and weapon systems.

C. MODELING COORDINATION

Another VDT feature that had no straightforward correspondence to DDD/Experiment 8 was the option to set meetings among decision makers as a means to improve communication and reduce the rework. It seems that VDT measures the quality of communication among DM's by tracing the fraction of communications responded to and meetings attended. Bottom-line: the more meetings (coordination) attended, the better the performance. However, we found this feature apparently without direct correlation with DDD/Experiment 8, as there were no scheduled meetings but rather open communication links established among the decision makers. Moreover, in DDD extensive (real-time) coordination is a *detractor* to performance, in terms of additional workload and consequent time delays.

D. USING VDT PROBABILITIES TO MODEL THE LEVEL OF COMMUNICATION/COORDINATION, WORK DELAYS, AND REWORK

The behavior of the VDT model is determined probabilistically, and simulated stochastically. As we have briefly described in Chapter III, VDT, allows the user to set probabilities (information exchange, noise, functional error and project error probabilities), thereby introducing an element of error into the project that makes it closer to reality.

1. Modeling the Level of Communication

VDT uses the information exchange probability to measure the level of communication within the project between positions responsible for tasks linked by communications links. In DDD/Experiment 8 communication and coordination are driven by varying degrees of task interdependence. To model this using VDT, we planned to use the exchange probability rate by setting it to a higher value for incongruent projects and a lower value for congruent ones.

2. Modeling Work Delay and Rework

The probability of noise used in VDT is "to account for the effect of interruptions in the ordinary working day that take time away from doing the project tasks" (SimVision^{TM} Tutorial, 2003). This may find correspondence in DDD/Experiment 8 with the level/probability of occurrence of unpredicted tasks and defend tasks that could divert the DM from performing an ongoing task, causing delays or even rework.

Functional error probability - the probability that a task will fail and require rework, has no direct correspondence in DDD/Experiment 8; it does not seem useful to incorporate rework links between interdependent tasks as long as the mission goes ahead without rework, and the probability of a weapon failure once launched is close to zero. However, there could be many critical cases in which targets are not destroyed on the first try, additional sorties are flown, or additional clearing operations are conducted. (The DDD allows for such "functional error probability" although it was not utilized in Experiment 8.) Another factor that is implemented and measured in DDD is the total number of wasted attacks (NumWast) that occurs when a task is attacked when the task's prerequisites are not met.

The addition of rework links may significantly contribute to more realistic VDT simulation results. The same reasoning may apply in the more complex but not unrealistic instance that implies the project error probability – "the probability that a task will fail and generate rework for itself and all failure-dependent tasks, which are tasks connected to it by rework links" (SimVision™ Tutorial, 2003).

E. COMPARING VDT AND DDD OUTPUT

Repeated simulation of a VDT model yields a database of organizational behavior. The database contains expected outcomes related to the schedule, cost, and quality of each simulation experiment. VDT tracks the final schedule duration of the entire work process, as compared with the projected duration based on Critical Path Method (CPM) analysis (which assumes no errors, and no explicit communication or rework). VDT tracks the project cost based on the salaries of the employees, but does not include task-related costs. Also, VDT computes the percent of a task's work volume that is direct work, versus the percent spent in coordination and rework.

Using the DDD Post Processor V2.1 (a database application designed to analyze experimental data after each DDD experiment) a vast amount of DDD output data can be transformed into useful information that can be exported to other statistical packages for further analyses (Wong, Kleinman, 2002). The DDD results can be plotted, filtered or drilled-down and compared by dependent variables, scenarios, teams, or decision-makers. The summary data contain a large set of observed dependent variables (DV) including the average response time (time of request – time of report), the average latency (time between task arrival and attack), the total gains/losses based on scoring method, the total number of wasted attacks (i.e., prerequisites not met) etc.

F. MODELING DESIGN TRADE-OFFS

A key research question is whether the behaviors replicated by VDT can apply in the DDD/Experiment 8 environment. A primary tension in any modeling effort lies in how much of the observed world one includes or excludes. Through modeling a complex system is simplified and abstracted to its most relevant essentials, while still making useful predictions. The choices a model-builder makes ultimately determine the questions the model can answer. And one very important choice is the level of detail, or granularity, at which the organization and its work is represented (Fridsma, 2003).

In our research we were aware that the goal was not to have VDT models resemble DDD models. Rather the goal was to develop VDT models that could reflect key behaviors of military forces in combat. Furthermore, in terms of validating our VDT models, such validation was not supposed to be against DDD models, but rather against the results of Experiment 8, which included human decision makers playing the DDD scenarios. Therefore, we were not concerned if our VDT models did not structurally match DDD models, as long as the replicated VDT behaviors could reflect qualitatively those exhibited during Experiment 8.

However our efforts to find as many relevant correspondences as possible, and to ameliorate the differences between VDT and DDD/Experiment 8, were critical for assuring that the modeling process would exclude all irrelevant elements and also for gaining confidence in our predictions. Hence, the results of these comparisons were subsequently exploited in our attempt to build simple independent VDT models to capture the relevant DDD/Experiment 8 behaviors.

G. IDENTIFYING DDD KEY BEHAVIORS

The ultimate step here was to identify the DDD key behaviors and assess the way we could possibly replicate them in VDT. The selected DDD key behaviors are briefly described below:

- *Expendable assets* – concerns the characteristic of some DDD assets (e.g., missiles, fighter sorties) to get consumed after use.

- *Resource Scarcity* – There are more task requirements than there are resources available. If DM's use more assets on a task than required, they are not available for use elsewhere.

- *Task precedence/prerequisite* – This relates to the sequence that should be followed when achieving specific tasks (e.g., clearing a minefield before landing on a beach), which require prerequisite (or corequisite) actions (Kleinman et al., 2003).

- *Tasks Prioritization* – A DDD feature is the presence of conflicting task requirements that requires prioritization (e.g., shifting to a defensive task while an offensive is underway).

- *Time criticality* of tasks – There are time constraints in the performance of certain tasks. If that is time critical tasks are not achieved there are consequent casualties or assets lost (e.g. SAR tasks, destroying enemy' SCUD launchers).

- *Multiple skill requirement tasks* – Some DDD tasks require multiple skills to be completed (e.g., destruction of Command Center requires two units of Strike and one unit of SOF).

- *Coordination requirements* – There are DDD parallel processing tasks, where assets owned by different DM's are to be coordinated in time and space (e.g., a complex task that may require an air, naval and ground attack).

- *Geography* – This is an important and complex DDD feature implying coordinated actions of DM's using different assets from different locations.

At this stage, we were not entirely aware of what we could or could not replicate in VDT. However, as part of the modeling design process we planned to follow an iterative course of action allowing us to go back, reassess and refine the models to try to capture these behaviors using VDT. This process of replicating the selected behaviors building and refining the models is described in more detail in Chapter V.

V. MODELING WITH VDT

The aim of VDT research is to develop computational tools to analyze decision making and communication behavior within organizations as discussed in Chapter III. It is an excellent tool to model projects in the corporate world. The aim of this project is to explore its potential for modeling organizational behaviors in a military context as represented by DDD Experiment 8 (herein referred to as DDD), discussed in Chapter II.

In this chapter we narrate the steps taken towards fulfillment of our objective. The aim is to develop a VDT model that replicates the key behaviors identified in DDD experiments. The sections below summarize steps undertaken toward this end.

A. LEARNING VDT AND DDD

The first step in this project involved getting familiar with VDT. Towards this end, licensed copies of VDT were purchased and all the project members familiarized themselves with the software by reading the User Manual (SimVision™ User Guide) and working the tutorials supplied with the software. Also, to understand the theory behind VDT, the project members referred to various papers and articles published on the subject (Jin and Levitt, 1996; Kunz et al, 1998; Levitt, R, 1996, Nissen et al, 2003).

In parallel, team members familiarized themselves with the workings of DDD, in particular with A2C2 Experiment 8. Towards this end team members referred to published material (Diedrich et al, 2002, Kleinman et al, 2003, Diedrich et al, 2003, Kleinman D, 2001) and also obtained first hand knowledge by observing a demonstration of Experiment 8 in progress. Team members also attended a players briefing for Experiment 9 to develop an appreciation for the nature of complexities involved.

B. IDENTIFYING KEY MODEL BEHAVIORS

Our primary objective was to build a representative model in VDT that replicates the performance and behaviors in DDD. The DDD task and organization structures consist of many complex scenarios. The team decided that in order to build a complete model we need to identify and model simple but key behaviors in DDD that will provide the building blocks for the final model. These behaviors have been discussed earlier in Chapter IV Section IV.G. Summary descriptions of these behaviors are presented below.

1. Expendable Assets

In DDD, each weapon asset (e.g., missiles, TLAM's, and ABMs) gets consumed after use. On the contrary, in VDT, actors do not get "consumed". After they are finished with a task they move on to the next one. The critical question here was how to build a model where an actor is used only once.

2. Resource Scarcity

In DDD, each target has specific resource requirement and if players use more than the required assets (e.g., firing two missiles when one is required) against a target these assets are unavailable for use elsewhere. VDT, by design, does not limit use of actors. If a position has more than required actors then all of them will be used and the task will be completed in a shorter period of time. The critical question here was how to restrict the use of resources by actors such that a resource scarcity can be created.

3. Task Precedence/Prerequisite

In DDD, certain tasks were sequential wherein prerequisite tasks had to be completed before undertaking primary ones (e.g., destruction of Bridge prior to destroying NBW). The critical question here was how to structure the programs, projects and organizations such that the task structure replicates the DDD task graph. This could however be done easily in VDT by placing individual tasks in parallel/sequential with respect to other tasks.

4. Task Prioritization

DDD was setup in such a manner that there were conflicting task requirements and the players had to prioritize them (e.g., shifting to a higher priority defensive task while an offensive is underway). The critical question here was how to assign varying priorities to tasks such that the order of execution is correctly determined by VDT.

5. Time Criticality of Tasks

In DDD, there were certain time dependent critical tasks that had to be assigned higher priority in order to successfully complete the mission (e.g., defensive tasks). If a player did not defend then he would lose some assets. Moreover, to the player, these tasks appeared randomly. The critical question here was how to model such tasks within the overall task structure given the difficulty of modeling random tasks within VDT.

6. Multiple Skill Requirement Tasks

Some tasks within DDD, especially the ones in the divisional scenario required multiple skills to be completed (e.g., destruction of Command Center requires two units of Strike and one unit of SOF). This cannot be readily modeled in VDT since each task can have only one skill requirement. The critical question here was how to simplify the complex tasks by breaking them down into simple tasks.

7. Coordination Requirements

DDD required its players to coordinate before undertaking a task (e.g., tasks such as Command Center requiring multiple assets in the divisional task scenario therefore required coordination by several players). The critical question here was how to enforce coordination among actors in VDT.

8. Geography

Geography is an important consideration in DDD. There may be more than one player available with required assets for a task but if the target is outside one's weapon range, he/she could not undertake it. Moreover, attacks from different places required different time units to complete the task (e.g., missiles fired from two different locations would not reach the target simultaneously). The critical question here was how to create a scenario that takes into account the geographical aspect of DDD simulation within VDT models.

9. Departments and Staffing

Departments and staffing come into play when differentiating between functional and divisional organizations. The critical question here was how to create departments and staff people such that the resulting organization replicated the functional and divisional organizations in DDD.

C. MODELING INDIVIDUAL BEHAVIORS

Based on the knowledge gained from working the VDT tutorials, the team set about building simple independent models to replicate each of these behaviors mentioned above. These key behaviors were short listed based upon our understanding of the DDD scenarios and upon discussions between project members and our advisors. Successful replication of these behaviors was vital before we could attempt to build the final model since all of these behaviors would need to be included in it. These behaviors play a key role in determining the outcome in VDT experiments and hence were essential to our project. The models developed as part of this exercise are included as an electronic attachment to this report and the related file names have been indicated against each of the behaviors discussed below.

1. Expendable Assets (Expendable.vpm)

The team decided that the best way to model this behavior is to incorporate long meetings in the model in such a way as to commence the meeting after completion of a task and assign the related positions to the meeting. The meeting duration is set longer than the project duration to tie up that position until the completion of the project. The proposed model is a very low level model, in that each asset/weapon (e.g., missiles) is represented by a position. The model and its simulated result are shown below:

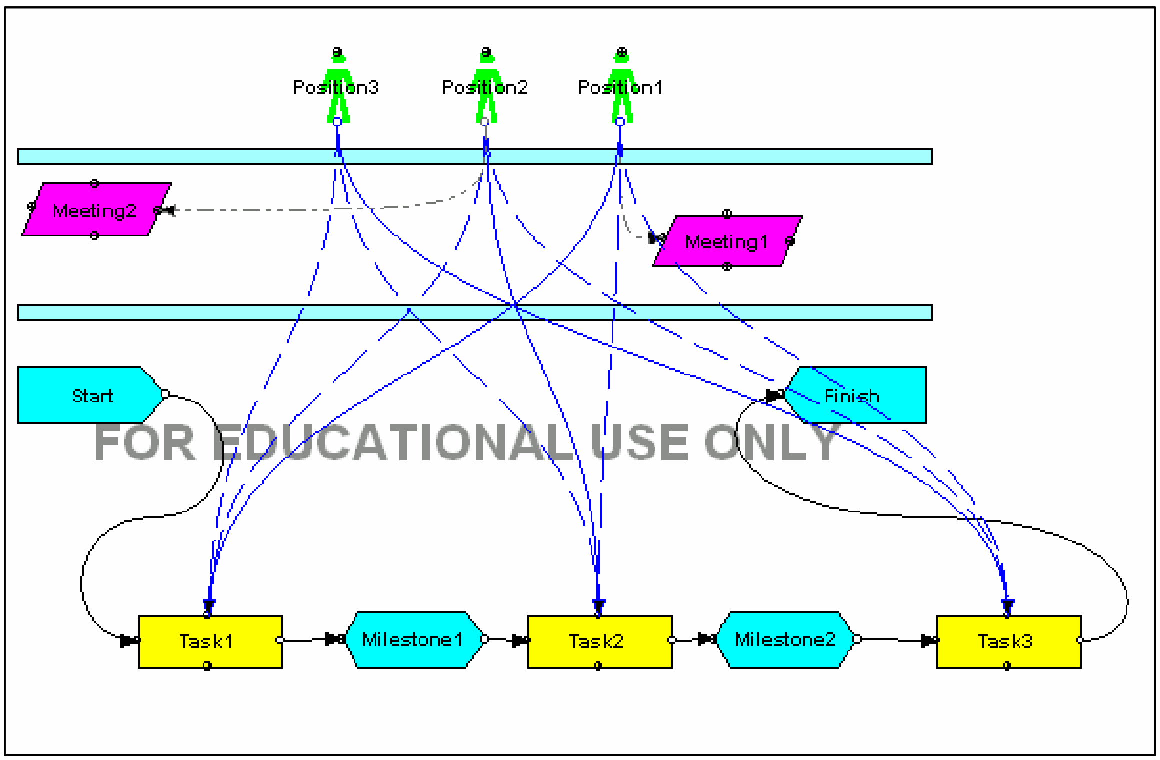


Figure 5.1 – Expending Assets

The model consists of three sequential tasks (Task 1, 2 and 3) and three positions (Position 1, 2 and 3). Figure 5.2 shows the task details. Each task is assigned to all three positions, however, only one position is assigned primary responsibility for each task (solid links) so that any position can undertake the task provided it is available. Two meetings (Meetings 1 and 2 related to Milestones 1 and 2 respectively) have been placed such that Position 1 is assigned to attend Meeting 1 and Position 2 is assigned to attend Meeting 2 (Figure 5.3). The aim was to make Position 1 unavailable after Task1 is completed. Similarly Position 2 would also be made unavailable after Task 2 is completed and only Position 3 would be available to execute Task 3. Although this could have been simply done by linking only one position to a task, it would fail to meet the requirement specified in Section V.C.3 that assumes positions with multiple task requirements.

Table View

Task	Name	Description	Priority	Work Type	Work Value	Units	Skills	Requirement Complexity	Solution Complexity	Uncertainty	Fixed Cost
1	Task1		Medium	Max Duration	16	Days	Generic	Medium	Medium	Medium	0
2	Task2		Medium	Max Duration	20	Days	Generic	Medium	Medium	Medium	0
3	Task3		Medium	Max Duration	22	Days	Generic	Medium	Medium	Medium	0

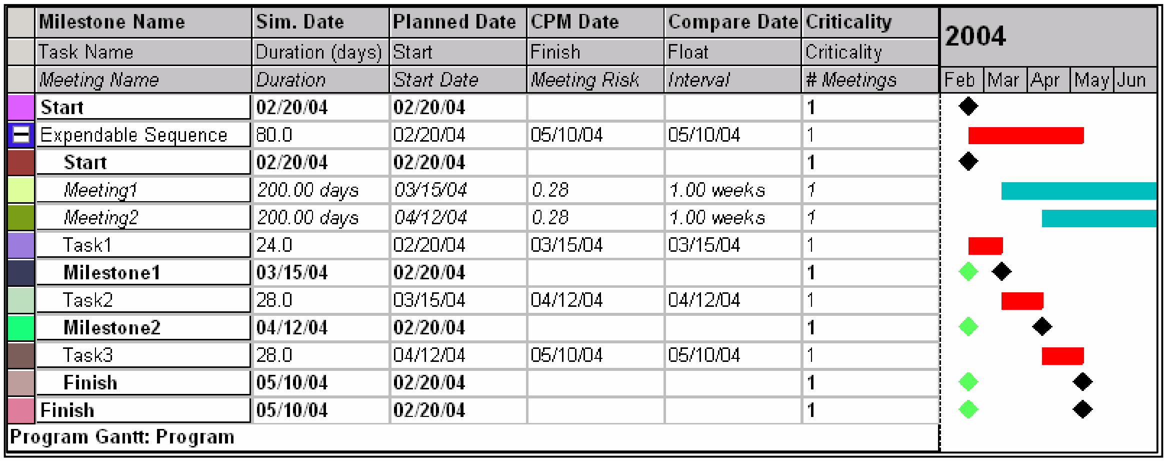
Figure 5.2 – Task Description

Table View

Meeting	Name	Description	Priority	Duration	Units	Repeating	Meet Every	Units	Start Time	First Meeting	Rel/Abs	Start Lag	Units	Schedule till
1	Meeting1		Medium	200	Days	No	1	Weeks	08:00 AM	Milestone1	Relative	0	Days	Till End
2	Meeting2		Medium	200	Days	No	1	Weeks	08:00 AM	Milestone2	Relative	0	Days	Till End

Figure 5.3 – Meeting Details

The simulated result for the above case is shown in Figure 5.4 below:



Milestone Name	Sim. Date	Planned Date	CPM Date	Compare Date	Criticality
Task Name	Duration (days)	Start	Finish	Float	Criticality
Meeting Name	*Duration*	*Start Date*	*Meeting Risk*	*Interval*	*# Meetings*
Start	**02/20/04**	**02/20/04**			**1**
Expendable Sequence	80.0	02/20/04	05/10/04	05/10/04	1
Start	**02/20/04**	**02/20/04**			**1**
Meeting1	*200.00 days*	*03/15/04*	*0.28*	*1.00 weeks*	*1*
Meeting2	*200.00 days*	*04/12/04*	*0.28*	*1.00 weeks*	*1*
Task1	24.0	02/20/04	03/15/04	03/15/04	1
Milestone1	**03/15/04**	**02/20/04**			**1**
Task2	28.0	03/15/04	04/12/04	04/12/04	1
Milestone2	**04/12/04**	**02/20/04**			**1**
Task3	28.0	04/12/04	05/10/04	05/10/04	1
Finish	**05/10/04**	**02/20/04**			**1**
Finish	**05/10/04**	**02/20/04**			**1**

Figure 5.4 – Simulated results (Gantt chart)

The Gantt chart above shows the duration, start and end times for various objects in the model. Tasks are represented by red rectangles, meetings are represented by blue rectangles and milestones are represented by rhombus. The Gantt chart above shows the effect of the meetings in the project. Position 1 is involved with Meeting 1 upon the completion of Task 1 and is not available for Tasks 2 and 3. Similarly Position 2 gets involved with Meeting 2 after completion of Task 2 and is not available for Task 3. This behavior can be further seen from the position backlog chart shown below.

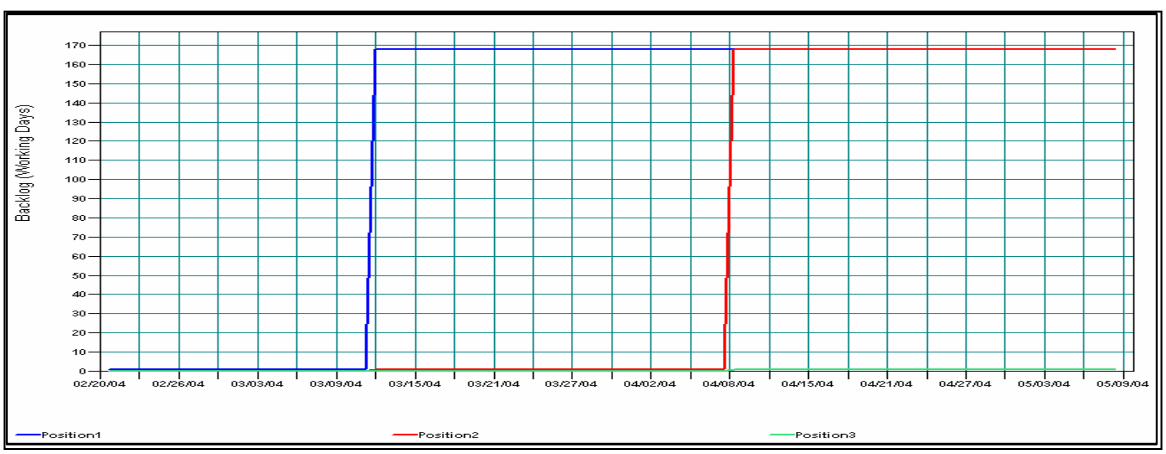


Figure 5.5 – Position Backlog Chart (Sequential Tasks)

The Position Backlog chart shows predicted overload of positions over time. The overload is expressed as a graph in terms of days for each of the positions in the project model with backlog measured along y-axis and days along x-axis. The above chart clearly represents the cycle of events taking place within the project. Position 1 attends Meeting 1 after completion of Task 1 and his work backlog increases close to 170 days at approximately 03/11/04 on the timeline and remains same thereafter. Similar is the case with Position 2 that attends Meeting 2 after completion of Task 2 that indicates a high work backlog beginning approximately 04/08/04. This indicates that positions 1 and 2 remain busy with meetings after completion of tasks 1 and 2 and are unavailable for further work until the end of project.

The above model could successfully replicate the behavior in DDD where assets such as missiles get consumed after use. However, this scenario was modeled at a very low organizational level with each position representing just one asset. A similar attempt at a higher organizational level involving departments and staffing is presented in Section V.3.8.

2. Task Precedence/Prerequisite

Task precedence can be easily modeled in VDT using sequential tasks. No separate model was built for this behavior. However, as can be seen from Figures 5.1 and 5.4, Task 2 commences only after completion of Task1, and Task 3 commences only after completion of Task2.

3. Task Prioritization (TaskPrioritization.vpm)

This behavior is directly related to DDD, in that, players are given limited assets and are faced with multiple tasks that may occur simultaneously. Some of these tasks are high priority tasks such as defensive tasks while some are low/medium priority tasks such as destroying SAM sites. The players may have to switch between tasks during the DDD simulation.

The team felt that the best way to model this behavior is to assign overlapping tasks (by incorporating varying delays for each task) to a position with limited FTE's and assign varying priorities to the tasks). Tasks are designed such that a high priority task commences while the low priority task is in progress.

The model was designed with four parallel tasks (SAM, DEF, CMD and SAR) with varying priorities assigned to a position (Position 1) with an FTE of one. All the tasks and the position have generic skill property. Various delays are incorporated in the tasks such that task SAM has no delay, task DEF has a delay of six days, task CMD has a delay of 10 days and task SAR has a delay of 30 days.

The model developed for this behavior is as shown in Figure 5.6, the various task properties are shown in Figure 5.7 and the various lags introduced before each task are shown in Figure 5.8.

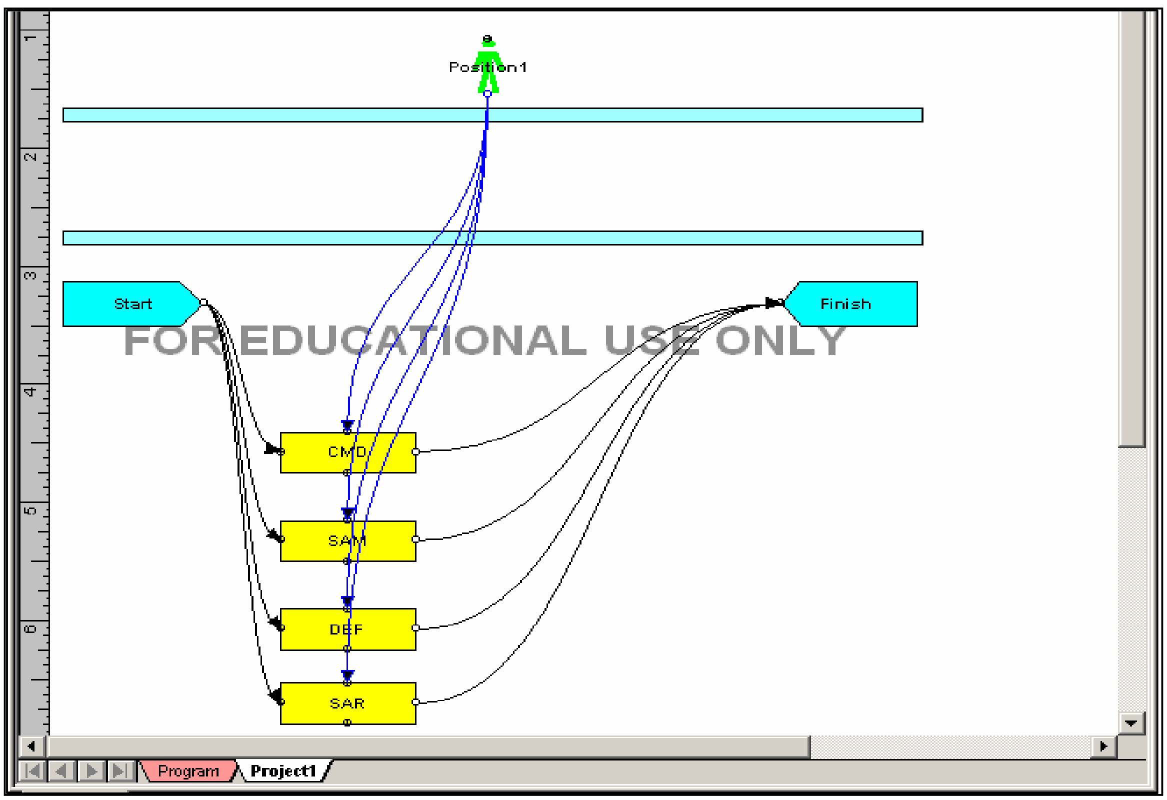


Figure 5.6 – Prioritization of tasks

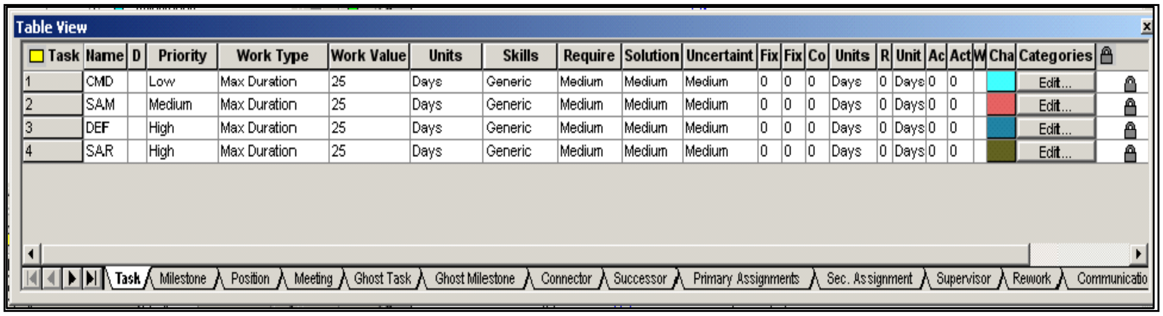


Table View

Task	Name	D	Priority	Work Type	Work Value	Units	Skills	Require	Solution	Uncertaint	Fix	Fix	Co	Units	R	Unit	Ac	Act	W	Cha	Categories	
1	CMD		Low	Max Duration	25	Days	Generic	Medium	Medium	Medium	0	0	0	Days	0	Days	0	0			Edit...	
2	SAM		Medium	Max Duration	25	Days	Generic	Medium	Medium	Medium	0	0	0	Days	0	Days	0	0			Edit...	
3	DEF		High	Max Duration	25	Days	Generic	Medium	Medium	Medium	0	0	0	Days	0	Days	0	0			Edit...	
4	SAR		High	Max Duration	25	Days	Generic	Medium	Medium	Medium	0	0	0	Days	0	Days	0	0			Edit...	

Figure 5.7 – Task details (note the varying priorities)

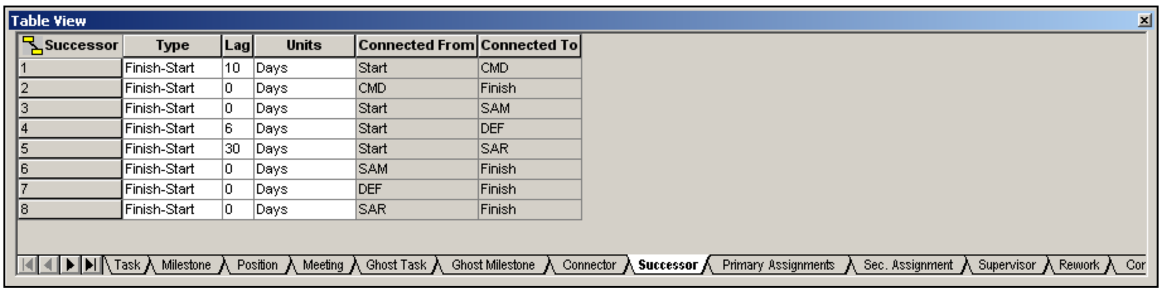
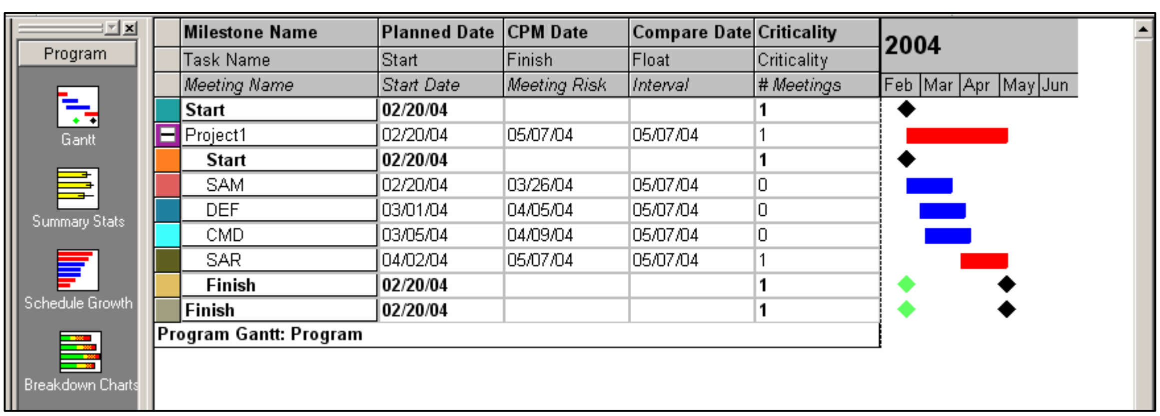


Table View

Successor	Type	Lag	Units	Connected From	Connected To
1	Finish-Start	10	Days	Start	CMD
2	Finish-Start	0	Days	CMD	Finish
3	Finish-Start	0	Days	Start	SAM
4	Finish-Start	6	Days	Start	DEF
5	Finish-Start	30	Days	Start	SAR
6	Finish-Start	0	Days	SAM	Finish
7	Finish-Start	0	Days	DEF	Finish
8	Finish-Start	0	Days	SAR	Finish

Figure 5.8 – Successor details (Notice the lags for the various tasks)

Following Figure 5.8, it was designated that the task SAM starts almost immediately, followed by the task DEF after six days. Position 1 should ideally leave or halt the task SAM in order to focus on DEF, it being a higher priority task. Similarly, after ten days, once task CMD is available, Position 1 should ignore it since it is already engaged in a higher priority task, DEF. The simulated result for the model in Figure 5.6 is shown in Figure 5.9.



Milestone Name	Planned Date	CPM Date	Compare Date	Criticality
Task Name	Start	Finish	Float	Criticality
Meeting Name	*Start Date*	*Meeting Risk*	*Interval*	*# Meetings*
Start	**02/20/04**			**1**
Project1	02/20/04	05/07/04	05/07/04	1
Start	**02/20/04**			**1**
SAM	02/20/04	03/26/04	05/07/04	0
DEF	03/01/04	04/05/04	05/07/04	0
CMD	03/05/04	04/09/04	05/07/04	0
SAR	04/02/04	05/07/04	05/07/04	1
Finish	**02/20/04**			**1**
Finish	**02/20/04**			**1**

Figure 5.9 – Simulated results (Gantt chart)

As can be observed from the above figure, the results do not match the assumptions made earlier. The Gantt chart shows that Position 1 undertakes all the tasks concurrently and finishes them in exactly the same order as specified in the model in Figure 5.6 (i.e., SAM – DEF – CMD – SAR). We can thus conclude that this model does not replicate the behavior of prioritizing and shifting task focus as required in DDD.

4. Time Criticality of Tasks (TimeCriticalityofTasks.vpm)

In DDD, certain defensive tasks are time critical and have higher priority than others. The players are expected to halt other operations and take on these tasks at the earliest. VDT does not provide any facility to specify time dependency for tasks (i.e., we cannot enforce VDT to complete a task within a specific time period). Tasks may only be placed in parallel or in sequence with other tasks and are then executed as per their respective position within the model. The attempted workaround for this problem was to introduce certain high priority tasks and make them prerequisite for the normal mission tasks. We then added delay for these high priority tasks such that they occur in the VDT model at the same time as they 'appear' in DDD. For example, if an enemy destroyer 'appears' after 10 minutes of starting in DDD we can delay the related task in VDT by the same factor. A simple representation of the above is shown in Figure 5.10.

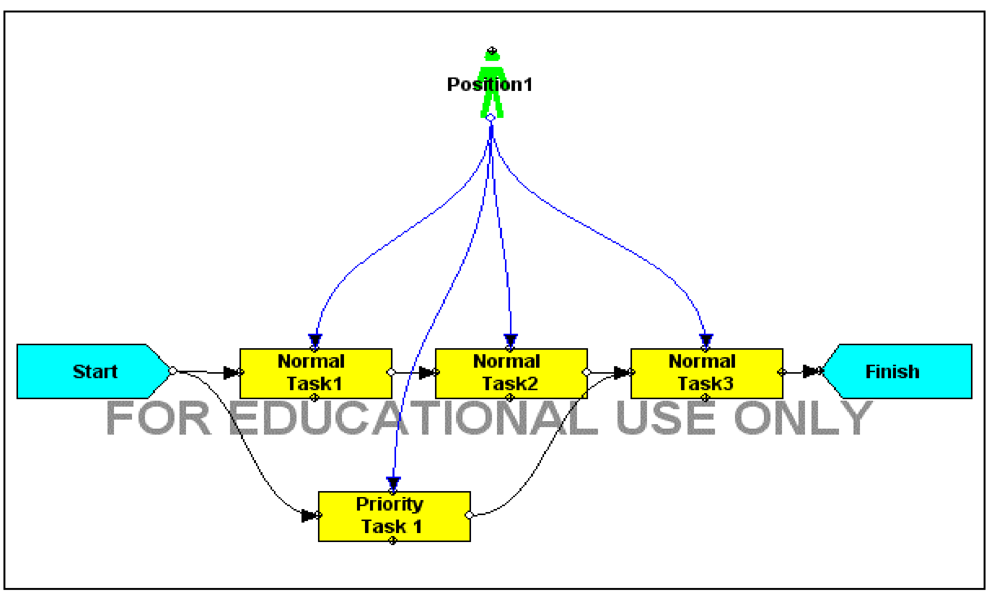


Figure 5.10 – Time Criticality of tasks

The above model consists of three sequential tasks with medium priority, generic skills and a work volume of 25 days. A fourth high priority task 'Priority Task 1' with generic skills and a work volume of 30 days is added. This task has been designed to start 35 days after 'Start' and is prerequisite for executing Task 3. This can be seen equivalent to DDD where the player continues executing primary tasks and is faced with a defensive task after some point in time (35 days here). Simulated result of the above model is shown in Figure 5.11.

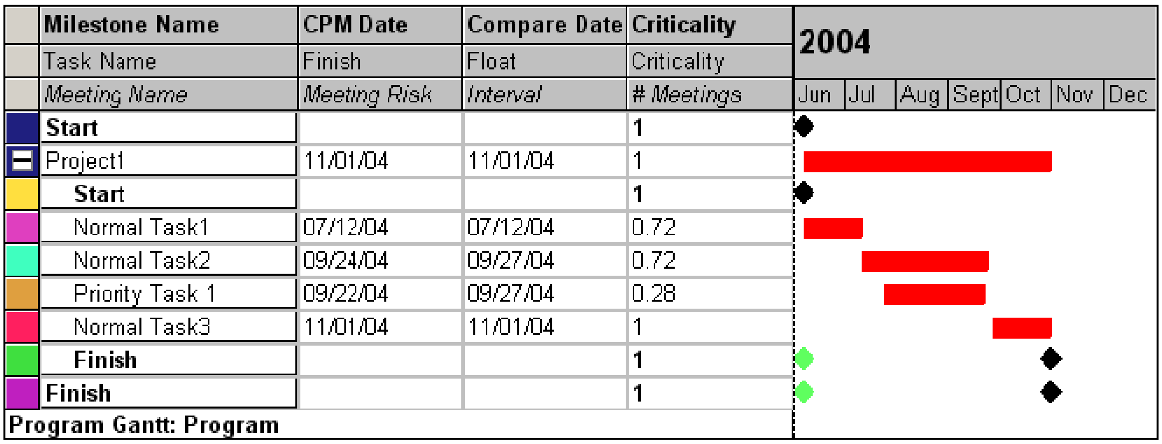


Figure 5.11 – Gantt chart for Time Criticality of tasks

The above result demonstrates the sequence of task execution. Normal Tasks 1, 2 and 3 are executed in sequence while the Priority Task 1 commences after 35 days and is finished before undertaking Task 3. Position 1 has to divert resources from task 2 and hence Task 2 takes longer than Normal Tasks 1 and 3.

5. Multiple Skill Requirement Tasks (Multiple Skill Requirement Tasks.vpm)

Some tasks in DDD within the Divisional task scenario require multiple skills in order to be executed (e.g., destruction of the Command Center requires two units of Strike and one unit of SOF). Since VDT allows us to specify only one skill requirement for any task, the problem was to define a model that can successfully replicate the required behavior.

The project team decided that the best solution would be to break down multiple skill requirement tasks into simpler tasks, each requiring just one skill. These tasks are placed in parallel such that all have to be executed simultaneously. The added advantage of this solution is that communication links may be placed between these tasks and that is helpful in modeling the coordination required between responsible positions in order to execute these tasks. A model developed for the above construct is shown in Figure 5.12.

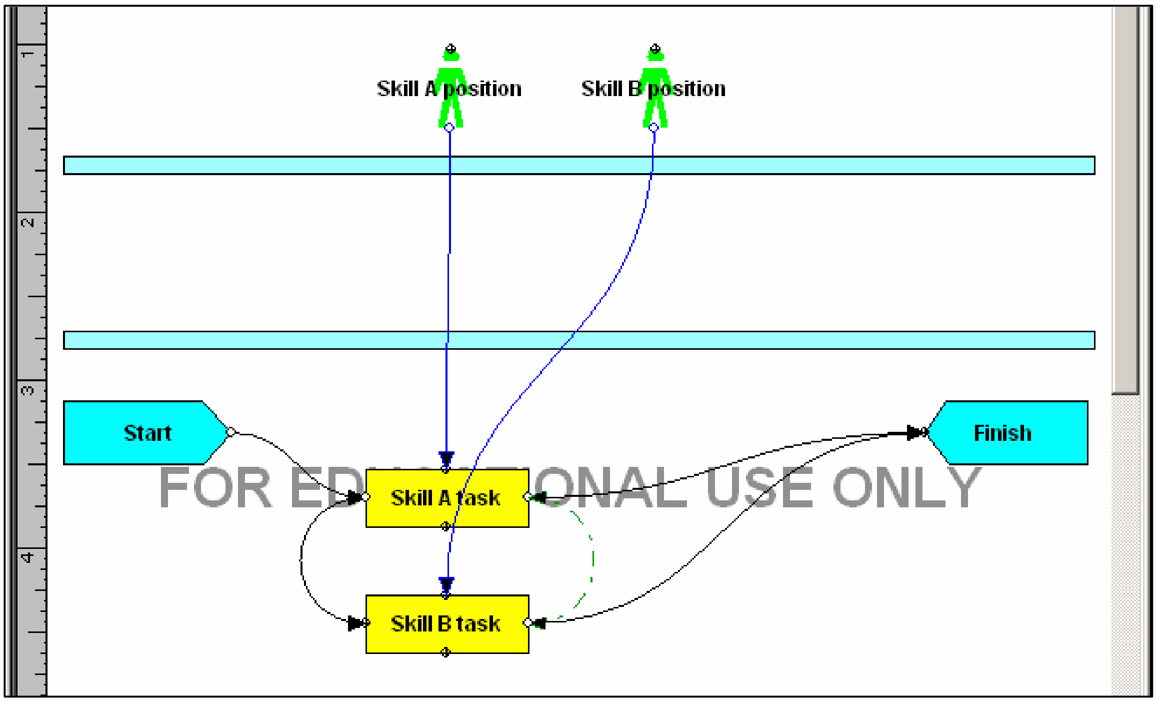


Figure 5.12 – Model representing breakdown of complex tasks

The above figure shows a representation of a single complex task that requires two skills, A and B, for successful execution. For example, the task set resembles the task CMD CTR in the DDD divisional task structure that requires two skills, Strike and SOF, to destroy it. The positions represent the players (say Strike and SOF commanders) such that each position is responsible for his/her part of the task set and need to coordinate with the other in order to execute the task successfully. Coordination is designated by the dotted line linking the two tasks.

Two skills have been established for the above model. While Skill A task and Skill A Position requires/have Skill A, Skill B task and Skill B position requires/have Skill B. The aim of this exercise was building a simple and intuitive model that would allow us to replicate multiple skill requirement tasks and would serve as building blocks for the final model. The simulated results for the model will not be presented here, as they do not mean anything in this isolated context.

6. Coordination Requirements

VDT provides a tool to model coordination among positions in the form of communication links. Setting the values of Information Exchange Probability specifies the severity of coordination required. The higher the value, the higher is the coordination required. No independent model was developed to model this behavior. However, as depicted in Figure 5.12 and explained in Section 5.3.5 above, communication links will be placed between parallel tasks and realistic values set for the Information Exchange Probability to model coordination requirements in the final model.

7. Geography (Geography.vpm)

Geography is one of the most important considerations in DDD. Platforms and assets can execute tasks only within their range of operations (e.g., a carrier can operate aircraft or launch missiles against targets only within its area of operations). Also, a SOF unit deployed in a particular region cannot undertake tasks in some other region. To do that, the units will have to be redeployed to the new region at the expense of time. One more important consideration is the timing of asset deployment to tasks (i.e., two missiles launched at a target from geographically different platforms at the same time would take different amounts of time to reach the target). There is also a window of operation limitation within the DDD that is related to the geography (i.e., the second missile has to reach the target within a certain time period of the first in order to completely destroy the target).

VDT does not offer a readymade solution to this problem. In VDT, models are representative of a process and the physical layout of positions and tasks themselves are not important. The project team tried various workarounds but we were not successful in modeling this aspect of DDD simulation. There were many different approaches attempted and it is not possible to include all of them here. One such attempt is presented below. As noted above, all models are placed as an electronic attachment to this document.

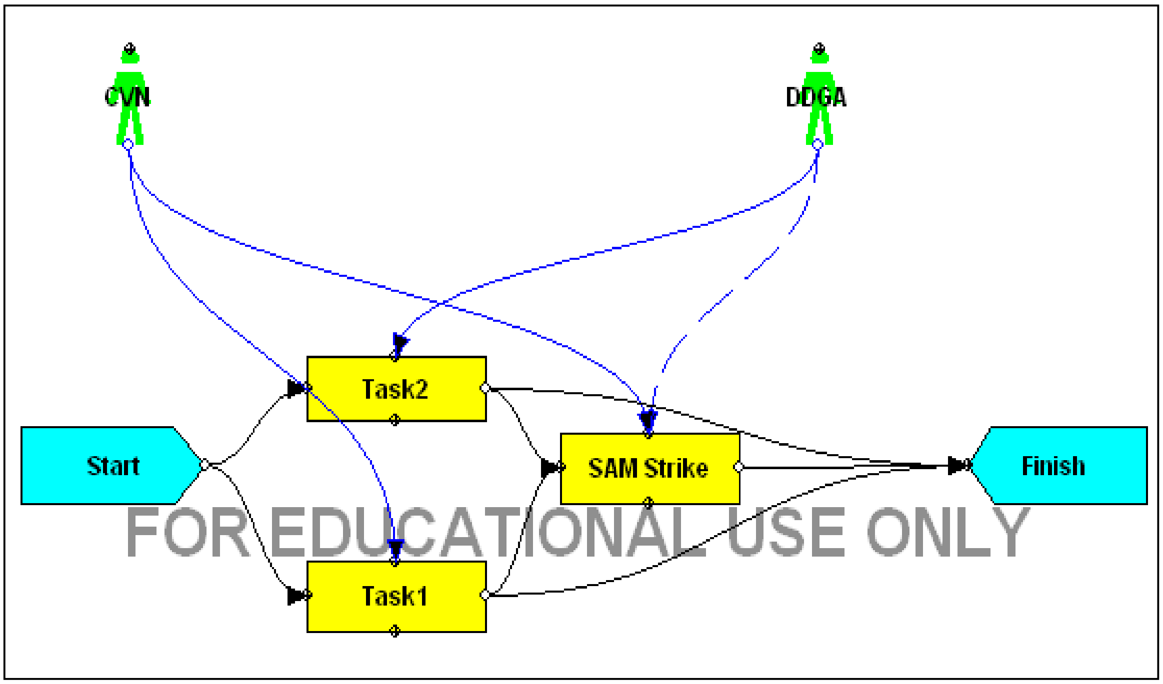


Figure 5.13 – Geography

The above figure represents an attempt to replicate geography in VDT. The model consists of one main objective – to destroy the SAM site. There are two positions deployed in different geographical locations, capable of doing the job. The SAM has a work volume of one day and will be destroyed by the first missile to hit it, i.e., when any of the positions execute it. Tasks 1 and 2 have been introduced to include an element of delay in the process and have been placed as a prerequisite for destroying SAM. Task 1 has a work volume of 15 units and Task 2 has a work volume of 10 units. This delay would suggest that a missile launched by CVN would take 15 units of time to reach SAM while it would take 10 units when launched from DDGA. Both positions CVN and DDGA have a FTE of one. Simulated result of the above model is shown in Figure 5.14.

Milestone Name	Planned Date	CPM Date	Compare Date	Criticality
Task Name	Start	Finish	Float	Criticality
Meeting Name	*Start Date*	*Meeting Risk*	*Interval*	*# Meetings*
Start	**02/20/04**			**1**
Project1	02/20/04	03/15/04	03/15/04	1
Start	**02/20/04**			**1**
Task2	02/20/04	03/05/04	03/12/04	0
Task1	02/20/04	03/12/04	03/12/04	1
SAM Strike	03/12/04	03/15/04	03/15/04	1
Finish	**02/20/04**			**1**
Finish	**02/20/04**			**1**
Program Gantt: Program				

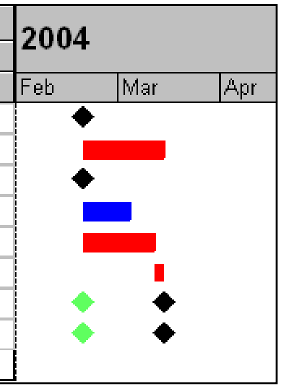


Figure 5.14 – Gantt chart for Geography

We can observe from the above figure that the task SAM is held up until both Tasks 1 and 2 are finished. According to our assumptions, it should have been executed immediately after completion of Task 2 but was not so. This would mean that if two missiles are fired at the target then the target would be destroyed only when both the missiles reach it. This is different from DDD wherein even if one weapon reaches the target we get a partial success. In VDT we were not able to achieve partial success. As can be seen above, the target would be destroyed only when all weapons (assets) reach the target Thus, this approach was not implementable in our final product.

8. Departments and Staffing (Expendable.vpm, Case - 'higher level')

An important result of DDD experiments pertains to behavioral differences between congruent and incongruent organizational structures and task structures. This was to be attempted in VDT models by creating two types of organizations, namely Functional and Divisional, and two types of mission task structures, namely those expected to favor either the functional or divisional organization. The team realized that to capture the unique behaviors represented by these structures, the models would have to be built at a higher organizational level than shown and described above for expendable assets (Section V.C.1). If each position were to represent just one asset, there would not be a major difference in behavior between the performances of Functional organization versus the Divisional organization (See the comparison between a functional and a lower level divisional model in Section V.F.12). This is because our project aims to model decision making regarding task assignments by the positions and this cannot be done if the positions represent just one asset each.

The basic idea is the same as presented in Section V.C.1, but positions now represent platforms (with multiple assets) instead of the individual/single assets. Three departments were defined containing thirty persons (ten each) and the positions were staffed with persons from the departments. The model is shown in Fig 5.15.

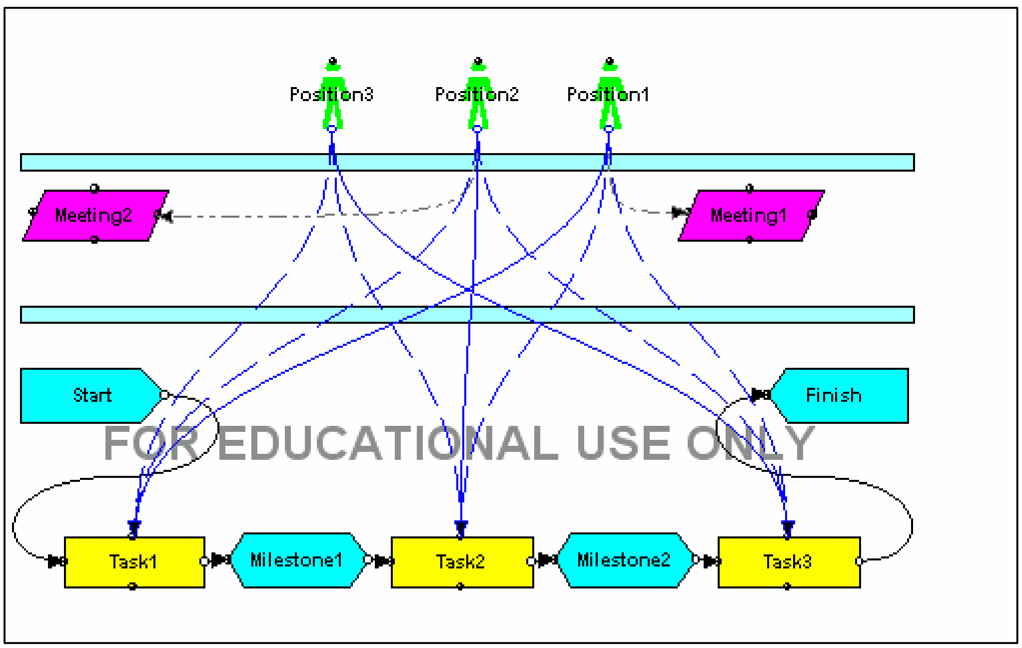


Figure 5.15 – Departments and Staffing

The above figure looks exactly like Fig. 5.1. However, the difference lies in the properties of tasks, positions and meetings. The new parameters are shown in figures 5.16, 5.17 and 5.18.

Task	Name	Description	Priority	Work Type	Work Value	Units	Skills	Requirement Complexity	Solution Complexity	Uncertainty	Fixed Cost	Fixed Rev
1	Task1		Medium	Max Duration	160	Days	Generic	Medium	Medium	Medium	0	0
2	Task2		Medium	Max Duration	200	Days	Generic	Medium	Medium	Medium	0	0
3	Task3		Medium	Max Duration	220	Days	Generic	Medium	Medium	Medium	0	0

Figure 5.16 – Task parameters for the model in Fig 5.15

All the task properties have remained the same except the work value. This property has been increased by a factor of ten to take into account the increased position FTE, which has been increased to reflect multi asset staffing.

Meeting Participant	Allocation	Connected From	Connected To
1	10	Position1	Meeting1
2	10	Position2	Meeting2

Figure 5.17 – Meeting participant's parameters for model in Fig 5.15

The meeting parameters have essentially remain the same. The only difference lies in the meeting allocation property. Since there are ten actors in every position, we wanted only one of them to attend the meetings so that the others are still available for the remaining tasks. Thus a meeting allocation of 10 percent has been defined that should theoretically restrict the attendance to the sub team leader.

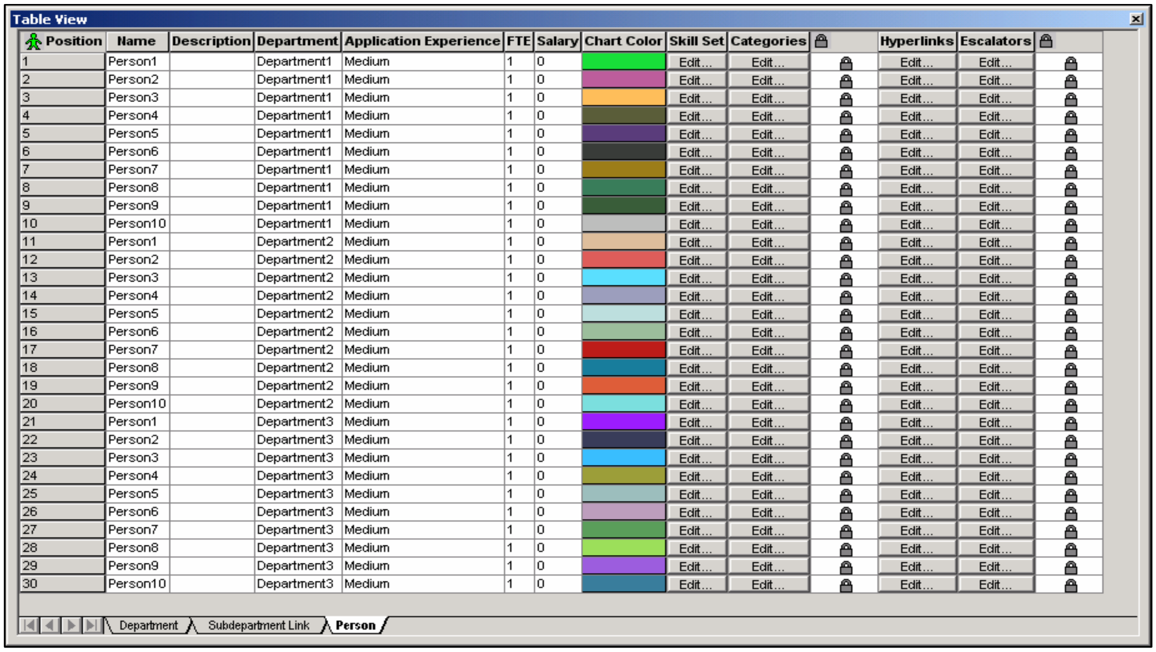


Table View

Position	Name	Description	Department	Application Experience	FTE	Salary	Chart Color	Skill Set	Categories		Hyperlinks	Escalators	
1	Person1		Department1	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
2	Person2		Department1	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
3	Person3		Department1	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
4	Person4		Department1	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
5	Person5		Department1	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
6	Person6		Department1	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
7	Person7		Department1	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
8	Person8		Department1	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
9	Person9		Department1	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
10	Person10		Department1	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
11	Person1		Department2	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
12	Person2		Department2	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
13	Person3		Department2	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
14	Person4		Department2	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
15	Person5		Department2	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
16	Person6		Department2	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
17	Person7		Department2	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
18	Person8		Department2	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
19	Person9		Department2	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
20	Person10		Department2	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
21	Person1		Department3	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
22	Person2		Department3	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
23	Person3		Department3	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
24	Person4		Department3	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
25	Person5		Department3	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
26	Person6		Department3	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
27	Person7		Department3	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
28	Person8		Department3	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
29	Person9		Department3	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
30	Person10		Department3	Medium	1	0		Edit...	Edit...		Edit...	Edit...	

Department | Subdepartment Link | Person

Figure 5.18 – Person list for model in Fig 5.15

Creation of departments and staffing is a new element in this model. Three departments have been created, each staffed with ten people as can be seen from Figure 5.18. All these actors have generic skills and are well suited to undertake any of the three tasks. All actors from Department 1 are staffed in Position 1 and so on.

In short, three major differences can be observed from the model in Fig 5.1. First, three departments have been created, one for each position, each containing ten persons. Positions have been staffed from their respective departments. FTE's for positions and task durations both have been increased by a factor of ten. A meeting participation allocation has been defined at ten percent, such that after completion of Task 1, only the team leader from Position 1 should attend the meeting and the rest can continue with other tasks. Our assumption is that the simulated results of the two scenarios (Figures 5.1 and 5.15) should exhibit similar results. The simulated results for the above model are shown below:

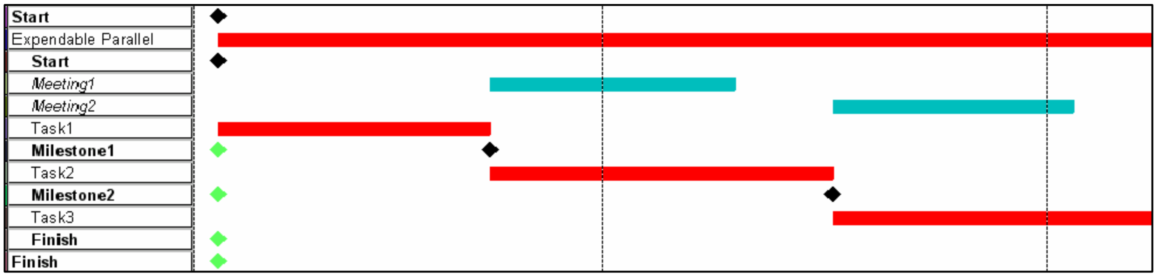


Figure 5.19 – Gantt chart for model in Fig 5.15

Figure 5.19 displays the Gantt chart for the model illustrated in Figure 5.15. The project team members observed an unexpected behavior. According to our assumptions, the task durations should ideally have been close to the ones observed in Figure 5.4 but was not so. In Figure 5.19, the tasks seem to take much longer to complete. Moreover, Tasks 2 and 3 take much longer than Task 1 and even surpass the meeting duration, which was set at 200 days. To further analyze the problem, we compare the project statistics between the two models (Figure 5.1 and Figure 5.15) in Table 5.1.

Case	**Lower Level**	**Higher level**
	Fig 5.1	Fig 5.15
Simulated Duration (days)	58	580
CPM Duration (days)	58	580
Simulated Start Date	2/20/2004	2/20/2004
Simulated Finish Date	5/10/2004	5/12/2006
Total Volume	346.0 days	7707.5 days
Work Volume	58.0 days	4987.5 days
Rework Volume	0.0 days	0.0 days
Coordination Volume	288.0 days	2720.0 days
Decision Wait Volume	0.0 days	0.0 days
Coordination Risk	0.093	0.107
Communications Risk	0	0
Meeting Risk	0.28	0.32
Percent Staffing	0	333

Table 5.1 – Project statistics comparison

Table 5.1 exhibits an anomalous behavior that the project team was not able to explain in the absence of any knowledge regarding the internal behaviors programmed into VDT. Note the coordination volume. It has jumped sharply in the higher level model to 2720 days. Also, because of this high coordination required, task durations have increased drastically; as can be seen from Fig 5.19 above, Tasks 2 and 3 extend beyond Meetings 1 and 2. It seems that until the time meetings are in progress, all the work is stopped and then resumes after the meeting is over. This finding disconfirmed our assumption that the work would continue while one of the ten members was attending the meeting.

D. MODELING SUCCESSES AND LIMITATIONS

The team achieved partial success in attempting to model individual behaviors. While some of the behaviors such as task precedence, coordination requirements and department and staffing could be modeled successfully, others such as geography, expendable assets and task prioritization were unsuccessful. Workarounds were developed for two of the unsuccessful attempts (Time criticality of tasks and Geography).

Time criticality of tasks was to be achieved by placing high priority critical tasks within the overall task structure in accordance with the DDD framework such that they would be made prerequisite for conducting other normal tasks. The actors will have to expend resources in executing these tasks and therefore will be constrained in their normal tasks. One difference that remains is that while in DDD, the actors may chose not to destroy a SAM site in order to assign scarce resources elsewhere, in VDT, once modeled, the task will be executed.

Geography was to be simulated by assigning tasks to actors within their geographical regions in accordance with the DDD scenarios. Thus, while an actor may have the skill to undertake all the tasks, he/she will be assigned to only those tasks that lie within his/her region of influence.

To further our learning and modeling, the project team decided to proceed with building a best effort representative model for DDD within the constraints of VDT. We wanted to observe the dynamics of a complex model after all the projected solutions in Section V.C were included. Also, the team felt that it would be easier to identify and isolate behaviors associated with VDT that are identical or in contrast with DDD, given the overall perspective of a complex task structure.

E. BUILDING DDD REPRESENTATIVE MODEL

To address the most important DDD scenarios and task structures, four models were built within VDT. These four models represent the functional (f) and divisional (d) task scenarios each being executed by a functional (F) and a divisional (D) organization structure. Further details of these scenarios within DDD are available in Section II.F. These models have been included as an electronic file ‘Final Model.vpm’, where each scenario is represented by a case within the model.

The first task was to create a task structure based on DDD (refer to Figure 2.5). Complex tasks in the divisional task structure (d) were broken down into simple tasks as discussed in Section V.C.5 above. Targets (task sets) were placed in parallel/sequential depending upon their respective positions within the DDD framework. A task requiring one asset to complete (destroy) it was assigned a work volume of 1 day. Skill requirement was determined depending upon the assets required by a given task. All other properties were left as default.

Six departments were defined each representing a major platform, namely the CVN, the three DDG’s, FFG and the CG. These departments were staffed with assets (sub platforms/ weapons in DDD) as per allocation within DDD (Figure 2.6). The departmental structure is presented in Figure 5.20.

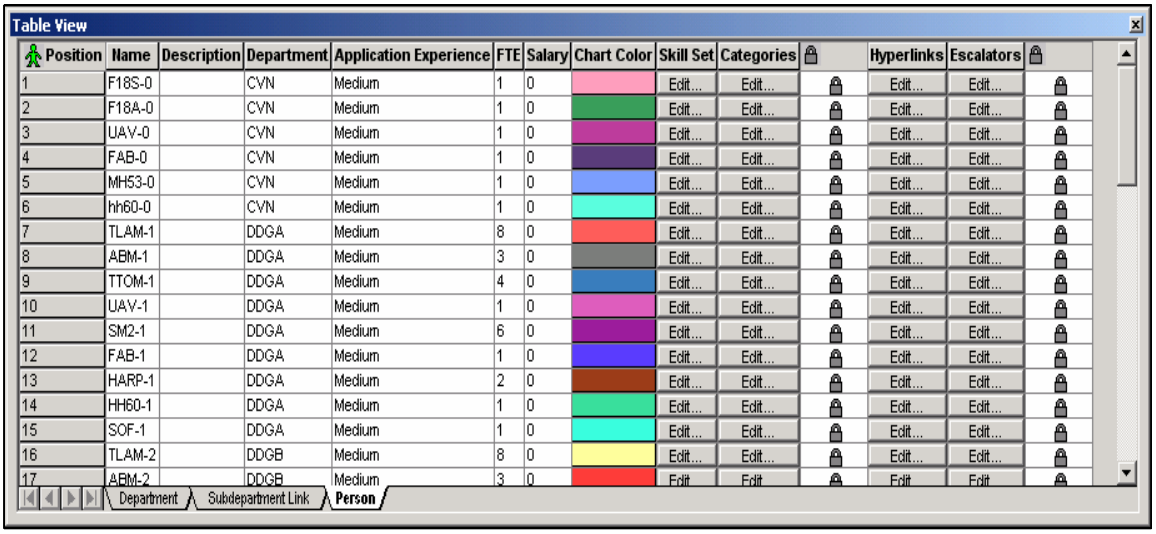


Table View

Position	Name	Description	Department	Application Experience	FTE	Salary	Chart Color	Skill Set	Categories		Hyperlinks	Escalators	
1	F18S-0		CVN	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
2	F18A-0		CVN	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
3	UAV-0		CVN	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
4	FAB-0		CVN	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
5	MH53-0		CVN	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
6	hh60-0		CVN	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
7	TLAM-1		DDGA	Medium	8	0		Edit...	Edit...		Edit...	Edit...	
8	ABM-1		DDGA	Medium	3	0		Edit...	Edit...		Edit...	Edit...	
9	TTOM-1		DDGA	Medium	4	0		Edit...	Edit...		Edit...	Edit...	
10	UAV-1		DDGA	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
11	SM2-1		DDGA	Medium	6	0		Edit...	Edit...		Edit...	Edit...	
12	FAB-1		DDGA	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
13	HARP-1		DDGA	Medium	2	0		Edit...	Edit...		Edit...	Edit...	
14	HH60-1		DDGA	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
15	SOF-1		DDGA	Medium	1	0		Edit...	Edit...		Edit...	Edit...	
16	TLAM-2		DDGB	Medium	8	0		Edit...	Edit...		Edit...	Edit...	
17	ABM-2		DDGB	Medium	3	0		Edit	Edit		Edit	Edit	

Department | Subdepartment Link | Person

Figure 5.20 – Staffing of personnel (assets) within departments.

Within the project window, six positions were defined each representing a human player in the DDD experiment. The nature of staffing of these positions would determine the organizational structure (F or D). In the functional organization all similarly skilled assets were allocated to one position. For example, the strike commander position was staffed with all strike assets from all departments and he was responsible for all strike related tasks. In the divisional organization each position was staffed with assets from a single department. For example, the CVN commander had all assets from the CVN department. These were multiple capability assets, and the commander was responsible for tasks within his/her region of influence. This difference in staffing can be seen from Figure 5.21.

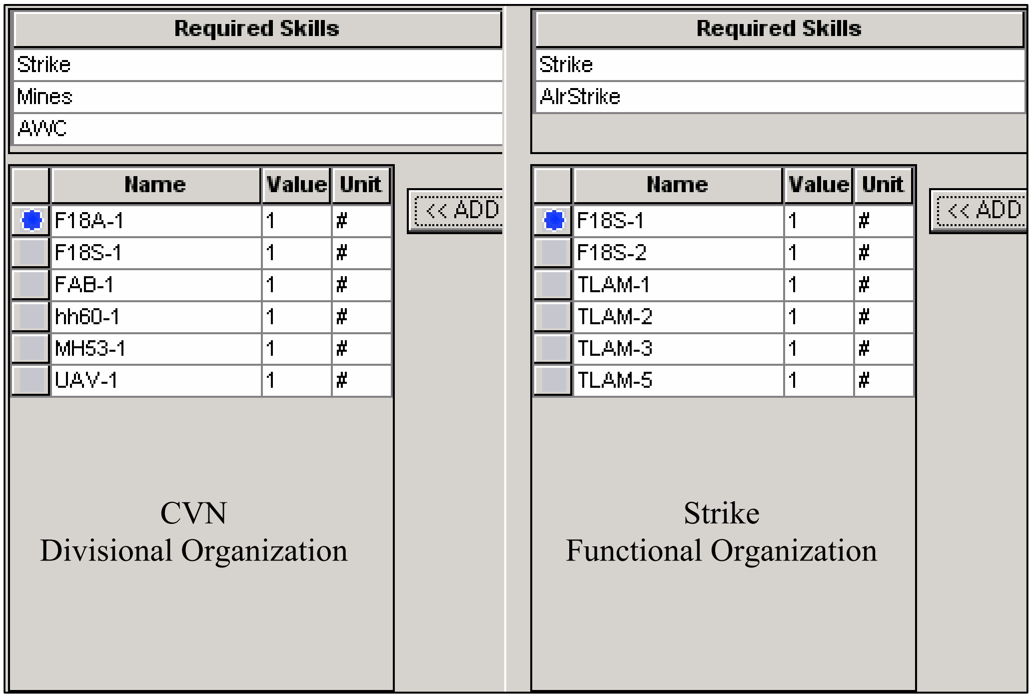


Figure 5.21 – Position staffing in the two scenarios

Due to our inability in modeling geography in VDT, the project team along with faculty advisors manually determined the tasks lying within a position's sphere of influence. For example, DDGA was tasked with targets in the western sector (ABW and NBW) since it was stationed in that sector in DDD (See Figure 2.4 in Chapter II).

One major problem was modeling task structures such as defensive and random tasks like SAM's. Within DDD, these are hidden from the players and only when an aircraft comes within its range (unknowingly), the SAM 'appears'. The player has to then redirect assets in order to destroy this SAM site before attempting primary targets (e.g., ABW and NBE). The project team decided to place five similar tasks (SAM) in sequence. These tasks would commence immediately after start and would be placed independent of the primary task structure. However, some of these tasks (exact number depending upon their placement in DDD) would be made prerequisite for undertaking primary tasks.

Communication links were placed between a set of parallel tasks. This was done in order to enforce coordination among the actors (players) responsible for the task. A value of 0.3, 0.1 and 0.1 was set for Information exchange probability, Noise probability and Functional error probability respectively. These values were set based on recommended and common levels in SimVision™ Users Guide (pages 85 – 89).

A skill set was defined that contained various skills required by the actors and is presented in Figure 5.22.

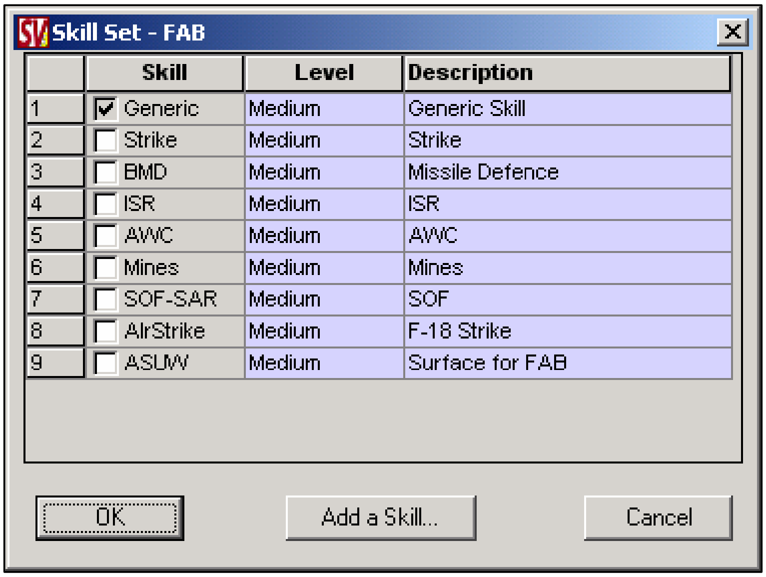


Figure 5.22 – Defined Skill set

Assets were assigned skills based on their role. For example, F-18S aircrafts has two skills 'AirStrike' and 'Strike' while TLAM's have only 'Strike'. This differentiates between tasks that can be undertaken only by aircrafts and not by missiles.

1. Divisional Organization and Divisional Task Structure - Case 'Divisional – Div'

A divisional organization was created by staffing positions from their respective departments (platforms). The six commanders (CVN, DDGA, DDGB, DDGC, FFG, and CG) were staffed with all actors from their equivalent departments (CVN, three DDG's, FFG and CG). Thus each position contained multiple assets of varying skills. Tasks were assigned to positions keeping in mind their geographical positioning within DDD. For example, DDGC was assigned primary responsibility for tasks in the eastern sector (ABE and NBE). Other positions in the sector provided secondary or support mission to it. The relevant model is shown in Figure 5.23.

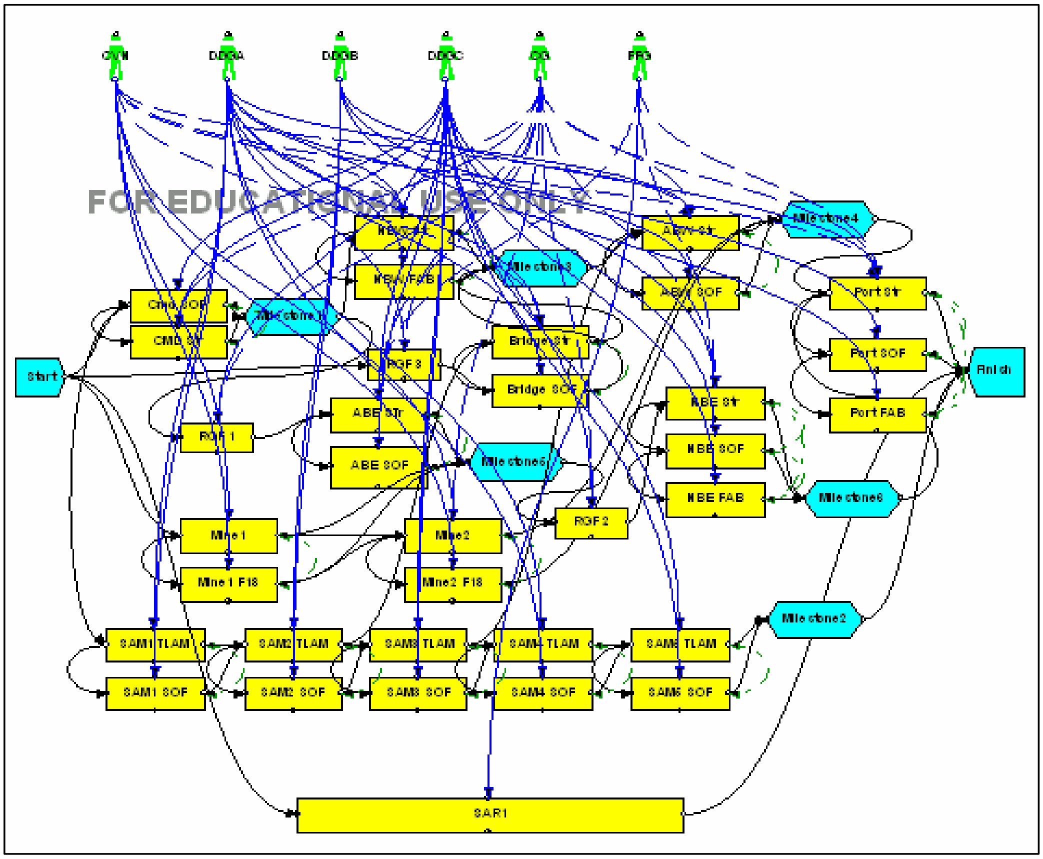


Figure 5.23 – Divisional – Divisional scenario (Congruent)

Note the structure of tasks in the figure above. They follow the task structure within DDD (Figure 2.5). However, there are differences. Two of the SAM (tasks) needs to be destroyed (completed) before attempting to destroy Bridge. Similarly, Mine 1 needs to be cleared before destroying NBE.

The simulated results of the above model are presented in section V.E.3.

2. Functional Organization and Divisional Task Structure– Case ‘Functional –Div’

The model for a functional organizational structure (F) with a divisional task structure (d) is shown in Figure 5.24.

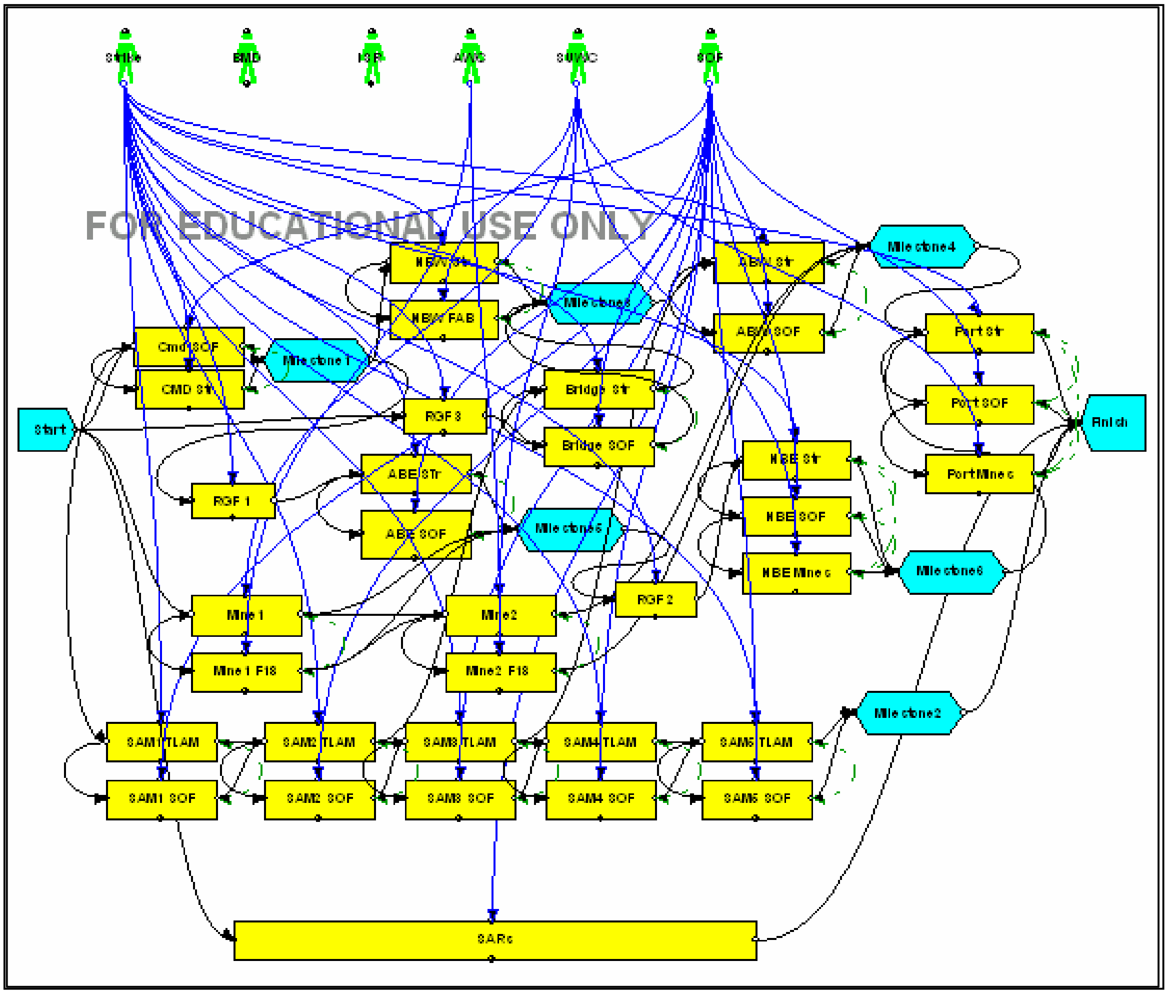


Figure 5.24 – Functional – Divisional scenario (incongruent)

The major difference between this scenario as compared to the one in Figure 5.23 is in the nature of staffing and task assignments. In this scenario, the six positions represent the functional commanders Strike, BMD, ISR, AWC, SUWC and SOF. The Strike commander is responsible for all strike related tasks, spread over the entire task structure. Other commanders are similarly responsible for their own set of tasks. This is unlike the congruent scenario (D-d) where the positions are responsible for tasks that lay within their sphere of influence (e.g., CVN commander was primarily responsible for Mine 1 and SAM 4 and provided support for all tasks in the eastern sector). This scenario (F-d) was designed in such a way that two or more positions (players) would be required to coordinate for completion of a task. For example, destruction of a SAM site requires coordination between the Strike commander and the SOF commander.

3. Comparison of Simulated Results

DDD Experiment 8 hypotheses that performance in the congruent (D-d) structure will be better than the incongruent structure (F-d). This is because in D-d tasks fall mostly under responsibility of one commander/position. In contrast, in F-d, tasks require assets across positions.

A comparison of the simulated results from the above two scenarios is shown below. The congruent scenario (d-D) is represented by the dark boxes, and the incongruent scenario (f-D) is represented by the hashed boxes. For each measure of interest, the dark box plots immediately above the hashed one.

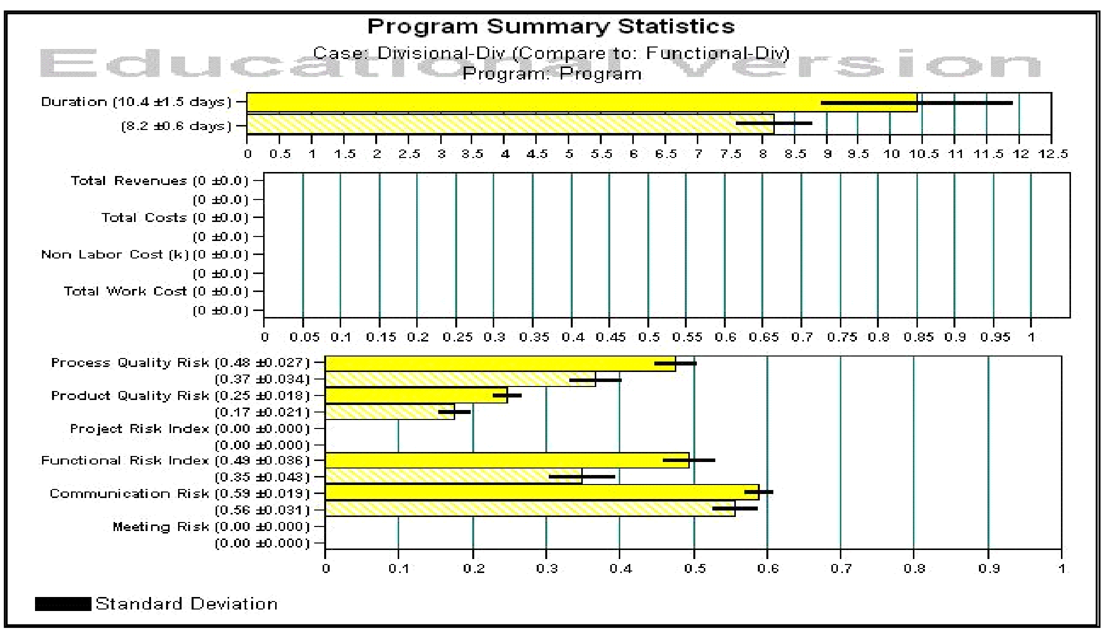


Figure 5.25 – Simulated result comparison between the scenarios

As can be observed from Figure 5.25, the results were exactly opposite to the assumptions based on Experiment 8. The incongruent scenario (F-d) seems to be more efficient than the congruent scenario (D-d) in all respects. This outcome is contrary to the results obtained from A2C2 experiments, in particular Experiment 8 (Section II.F).

In view of the contrary results obtained above, the project team began the task of identifying the causes of differences. A detailed task wise comparison is shown below. The congruent scenario (D-d) is represented by the dark boxes and the incongruent scenario (F-d) is represented by the hashed boxes.

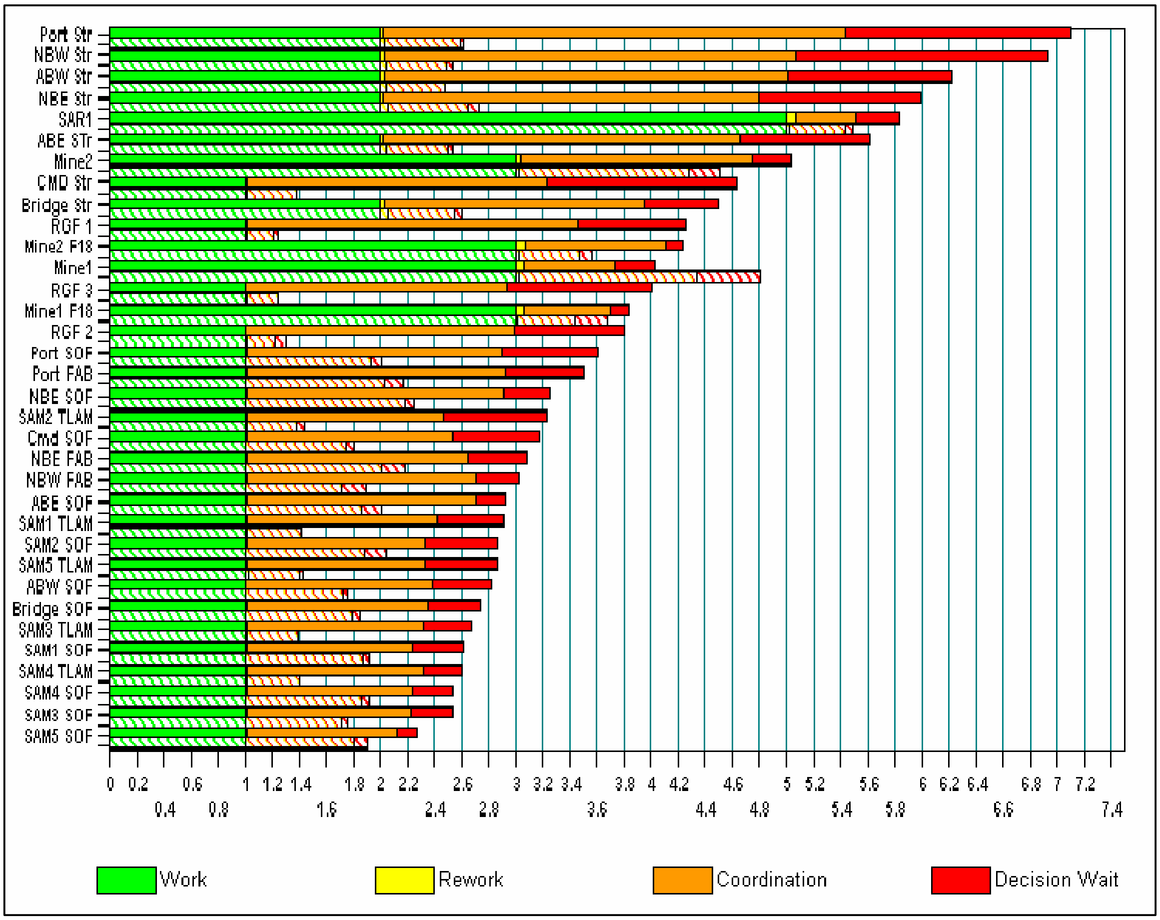


Figure 5.26 – Detailed task wise comparison between scenarios

As can be observed from Figure 5.26, the incongruent scenario (F-d) fares better at performing almost all of the tasks. The work volume is the same in both structures, which is expected, since both task structures defining the scenarios are identical. The rework volume is miniscule and may be neglected. However, the major difference is in the coordination and decision wait volumes. The above figure shows that coordination requirements and decision wait times are much higher in the divisional organization than in the functional organization. This is exactly opposite to our assumptions given that the task structure was designed to favor the divisional organization over the functional organization.

F. SIMPLE REPRESENTATIVE MODELS

Subsequent to the unexpected behaviors from the above models, the team decided that building further scenarios without resolving this issue was not warranted. Our approach then was to try and understand the behavioral nuances of the two scenarios by building simple representative models and studying them. These models were developed in an incremental fashion such that at every step we were able to either identify a particular behavior or isolate it in the context of our project. These models are included in the electronic attachment as 'A2C2 department behavior test.vpm'.

To study various behaviors and to rule out effects due to factors such as communication links the team set out to build models starting from a simple model of one position with two tasks where one task was suited to the skill of the position and the other was not. The aim was to see whether the incompatible behaviors observed between the functional and divisional scenario in Figures 5.23 and 5.24 could be replicated in relatively simpler representation.

1. Simple Model with One Position and Two Tasks

The first model (Case 'Baseline') consisted of one position with an FTE of one having Skill A. The position was responsible for two sequential tasks A and B (both with a work volume of 50 days). While Task A required Skill A (match with the position), Task B required Skill B (mismatch with the position). As expected, the results showed that completion of Task B requires more time than Task A (157 days as compared to 71.2 days). We changed the skill of the position to Skill B with the same task structure (Case 'Skill B Responsibility'). In this case the results were similar though Task A now required more time than task B (151.1 days as compared to 74 days). We also experimented with assigning both skills A and B to the position (Case 'S A& B') to observe the difference. As expected, both tasks require similar amount of time (71.3 days and 73 days).

2. Simple Model with Two Positions and Two Tasks

The second step involved adding one more position (Position B) with an FTE of one having Skill B. While Task A was assigned to Position A, Task B was assigned to Position B. We expected to see similar results between this model (Case 'SA SB') and the one in section 5.6.1 since the tasks were sequential. The results matched our predictions in that Task A took 71.2 days and Task B took 72.9 days.

3. Communication Link

The next step involved adding communication link between the tasks (Case 'SA SB + comm') with an Information exchange probability of 0.3, Noise error probability of 0.1, Functional error probability of 0.1 and Project error probability of 0.1. We felt that adding a communication link would increase the coordination requirement and thus the total workload but this was not so. The results indicate no appreciable change as compared to the model in Section V.F.2 (71.3 days for Task A and 72.9 days for Task B). We also tried the above configurations with parallel tasks (Case '+ concur' and '+concur comm') and obtained similar results.

4. Increased FTE and Work Volume

The next step was to proportionally increase the FTE's and the work volume by a factor of ten (Case '+concur x 10'). We wanted to see whether increasing the workload causes any changes with the system. The results (74 days for each task) showed that there was no appreciable change. This was expected since even the FTEs had been increased proportionally.

5. Adding Organizational Hierarchy

We wanted to study behavior changes caused by organizational hierarchy (Case '+con/con x 10 + boss'). To model this we added a sub team leader on top of the two Subteams (Positions A and B). We felt that vertical growth of organization would increase the workload due to increased coordination and decision wait times. However we found the results (74.1 and 74.2 days respectively) to be similar to those in Section V.F.4 above thus indicating that minor changes in organizational hierarchy alone do not cause appreciable changes in the amount of time required to complete tasks.

6. Departments and Staffing

We also wanted to study the behavior change due to introduction of departments and staffing (Case '+ org'). For this we created two departments, A and B, and staffed them with ten people each. People in department A had skill A, while people in department B had skill B. Positions A and B were staffed with ten people from departments A and B respectively. The results (70.3 days and 74 days) were same as the model in Section V.F.4 before. This indicated that there is no appreciable difference in performance between the two scenarios; one in which the unstaffed position has 10 FTE's and the other in which the position was staffed with 10 people.

7. Building representative models for Functional and Divisional Scenarios

We could establish from the models that, factors such as communication links, organizational hierarchy and staffing does not cause appreciable changes in project performance times. In this light, the team proceeded to model simple representative scenarios for the functional and divisional organizations in Figures 5.23 and 5.24. For this the team built four models. Three of them deal with different versions of a divisional organization and one with the functional organization model. The task structure is common across all four models.

The task structure basically consists of four individual tasks divided into two sets of two tasks each (Figure 5.27). There are two skills defined (Skill A and B) common to all the models. Of the two tasks in each set, one requires Skill A and the other Skill B. Each task has a work volume of 500 days. All the tasks are arranged in parallel and the tasks within each set are connected via communication links. The Information exchange probability has been set at 0.3, Noise error probability at 0.1, Functional error probability at 0.1 and Project error probability at 0.1. These are recommended and common values as per SimVision™ User Guide (pages 85 – 89).

There are two independent positions with a defined FTE of 10 (except in one scenario, Section V.F.11). The nature of staffing of these positions will determine the structure of the organization as well as provide us with inputs that are relevant to study the various aspects of organizational behavior.

There are two departments, 1 and 2. Each department is staffed with 10 people. Of the total 20 people, 10 have Skill A, and the rest have Skill B. The four models are discussed in detail below. Simulated results from all the models are discussed in Section V.F.12.

8. Divisional Organization (Case 'divisional')

In this scenario (Figure 5.27), Department 1 contains all ten people having Skill A, and Department B has all ten people with Skill B. Position 1 is staffed with five people from Department 1 (Skill A) and five people from Department 2 (Skill B). Position 2 is staffed with rest of the ten people (five each of both skills). Position 1 is tasked with Tasks 1 (Skill A) and 2 (Skill B) while Position 2 is tasked with Tasks 3

(Skill A) and 4 (Skill B). This organization thus represents two multi functional positions each that should be fully capable to meet the requirements of the two tasks assigned. This is a congruent divisional model (D-d).

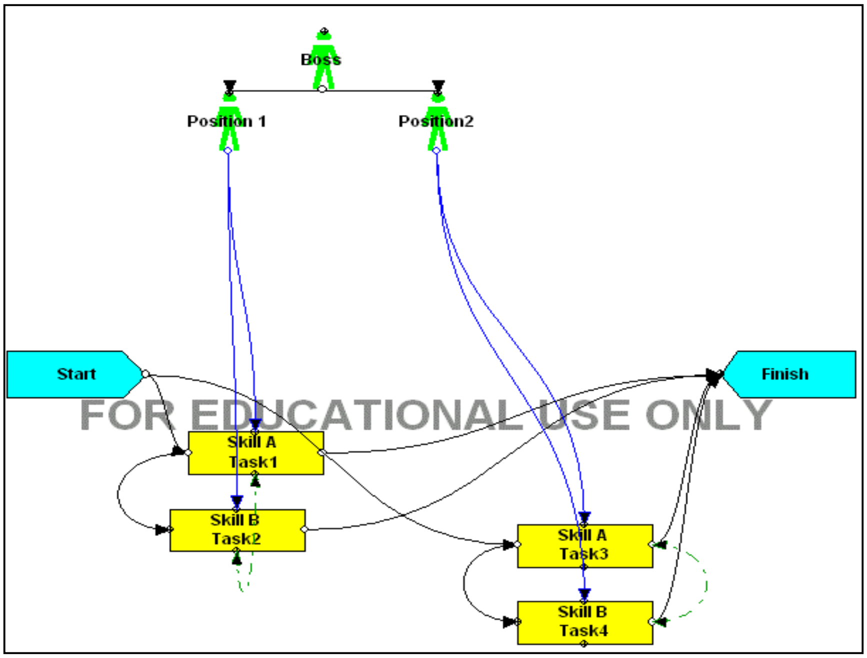


Figure 5.27 – Divisional scenario

The department organization and staffing is shown below.

Position	Name	Description	Department	Application Experience	FTE	Salary	Chart Color	Skill Set
1	Person1		Department1	Medium	1	0		Edit...
2	Person2		Department1	Medium	1	0		Edit...
3	Person3		Department1	Medium	1	0		Edit...
4	Person4		Department1	Medium	1	0		Edit...
5	Person5		Department1	Medium	1	0		Edit...
6	Person6		Department1	Medium	1	0		Edit...
7	Person7		Department1	Medium	1	0		Edit...
8	Person8		Department1	Medium	1	0		Edit...
9	Person9		Department1	Medium	1	0		Edit...
10	Person10		Department1	Medium	1	0		Edit...
11	Person1-2		Department2	Medium	1	0		Edit...
12	Person2-2		Department2	Medium	1	0		Edit...
13	Person3-2		Department2	Medium	1	0		Edit...
14	Person4-2		Department2	Medium	1	0		Edit...
15	Person5-2		Department2	Medium	1	0		Edit...
16	Person6-2		Department2	Medium	1	0		Edit...
17	Person7-2		Department2	Medium	1	0		Edit...
18	Person8-2		Department2	Medium	1	0		Edit...
19	Person9-2		Department2	Medium	1	0		Edit...

Figure 5.28 – Department organization: Divisional structure

Ten persons (Persons 1 to 10) belong in Department 1 and have Skill A while the other ten (Persons 1-2 to 10-2) belong to Department 2 and have Skill B. Positional staffing of these people can be seen in Figure 5.29.

Position 1

	Name	Value	Unit
	Person1	1	#
	Person2	1	#
	Person3	1	#
	Person4	1	#
	Person5	1	#
	Person1-2	1	#
	Person2-2	1	#
	Person3-2	1	#
	Person4-2	1	#
	Person5-2	1	#

Position 2

	Name	Value	Unit
	Person6	1	#
	Person7	1	#
	Person8	1	#
	Person9	1	#
	Person10	1	#
	Person10-2	1	#
	Person9-2	1	#
	Person8-2	1	#
	Person7-2	1	#
	Person6-2	1	#

Figure 5.29 – Positional staffing in divisional structure

9. Functional Organization (Case 'functional')

In this scenario (Figure 5.30), the departmental organization is same as in Section V.F.8. The difference lies in staffing of people. Position 1 is staffed with all ten people from Department 1 (Skill A) and is responsible for tasks 1A and 2A (both requiring Skill A). Position 2 is staffed with all ten people from Department 2 (Skill B) and is responsible for tasks 1B and 2B. This structure is similar to the one we have developed in Figure 5.24 wherein each player (position) is responsible for all tasks related to his/her skill throughout the task structure. For example, we can assume that Position 1 here relates to the Strike Commander in Figure 5.24 and he/she is responsible for all the Strike related tasks (represented here by Skill A).

In contrast with the model described in Section V.F.8 above, this represents an incongruent structure. The functional organization (single skill positions) requires coordination between positions to accomplish the simultaneous tasks. This required inter position coordination is hypothesized to generate increased workload based on greater information processing requirements (Galbraith and the results of DDD Experiment 8).

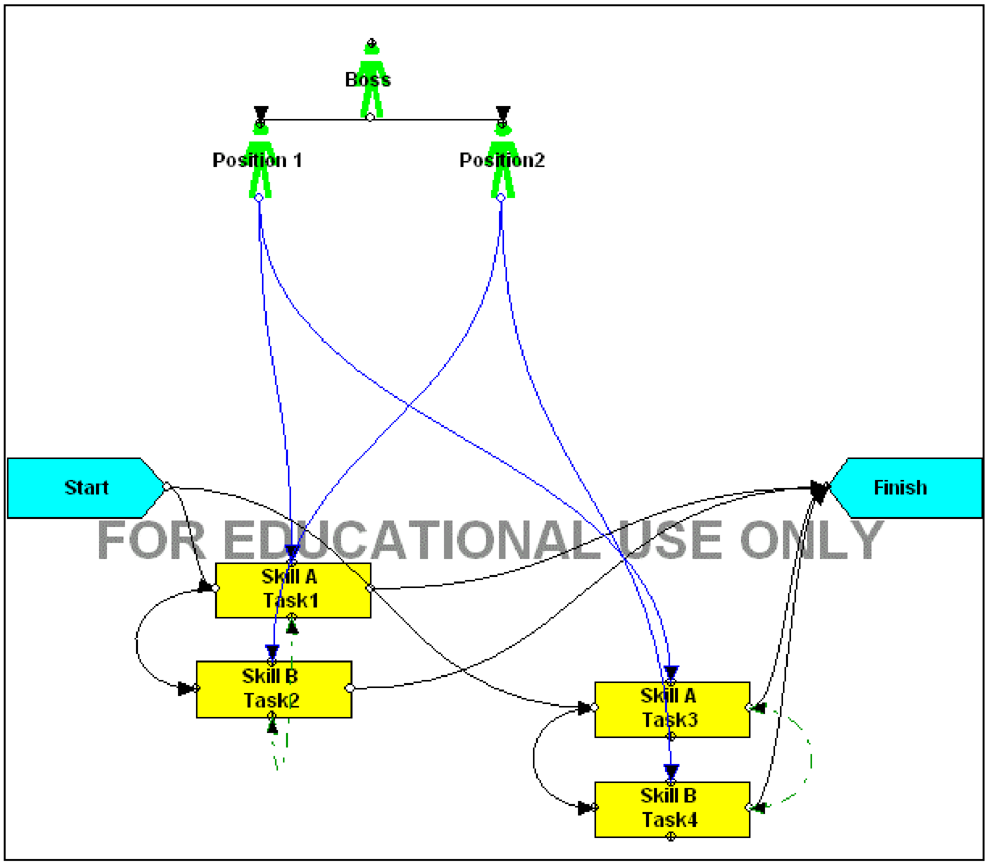


Figure 5.30 – Functional structure

The positional staffing for this scenario is presented in Figure 5.31 below.

Position 1

Name	Value	Unit
Person1	1	#
Person2	1	#
Person3	1	#
Person4	1	#
Person5	1	#
Person6	1	#
Person7	1	#
Person8	1	#
Person9	1	#
Person10	1	#

Position 2

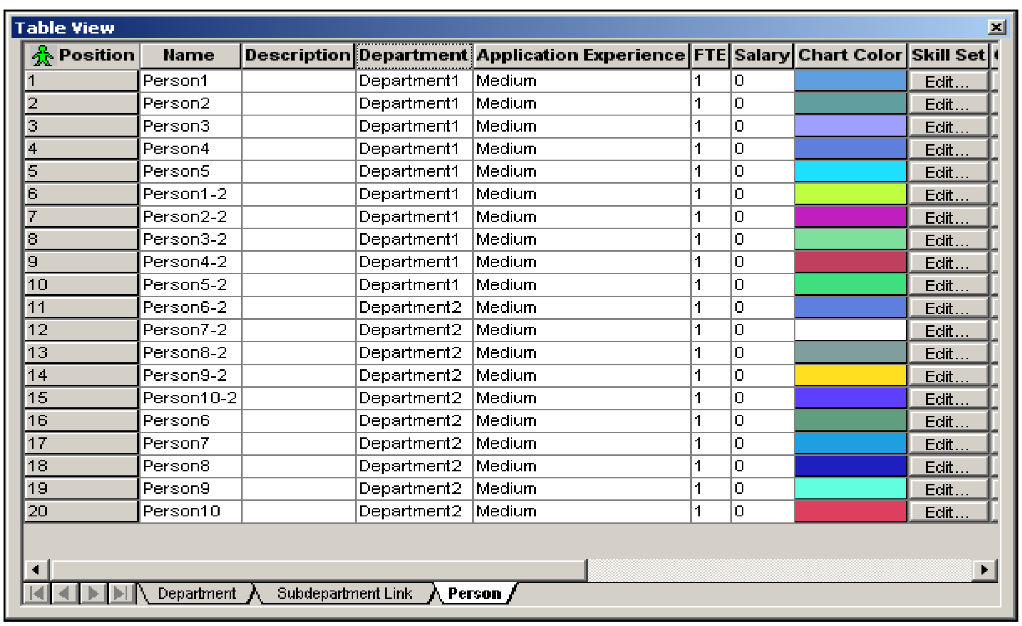
Name	Value	Unit
Person10-2	1	#
Person9-2	1	#
Person8-2	1	#
Person7-2	1	#
Person6-2	1	#
Person1-2	1	#
Person2-2	1	#
Person3-2	1	#
Person4-2	1	#
Person5-2	1	#

Figure 5.31 – Positional staffing in functional structure

10. Divisional Organization (Reversed Staffing) (Case 'Divisional – reversed staffing')

While in the functional scenario, all ten people from each department were staffed in one position, in the divisional scenario it was divided between the positions with five people from each department staffing each position. One primary concern of the project team was whether this was affecting the outcome. Thus the team decided to create a divisional scenario where staffing would be identical to the one in the functional structure.

In this structure, Department 1 is staffed with five people with Skill A and five people with Skill B. Department 2 is staffed with the other ten people (five each of Skill A and B) as can be seen in Figure 5.32. Position 1 is staffed with all ten people from Department 1 and Position 2 is staffed with all ten people from department 2. This model in comparison with that in Section V.F.8 has all staff for a position coming from one department. Thus, any increase in workload presumed by VDT because of interdepartmental staffing in Section V.F.8 should be eliminated here. The organizational and task structures are exactly the same as that in Figure 5.27.



Position	Name	Description	Department	Application Experience	FTE	Salary	Chart Color	Skill Set
1	Person1		Department1	Medium	1	0		Edit...
2	Person2		Department1	Medium	1	0		Edit...
3	Person3		Department1	Medium	1	0		Edit...
4	Person4		Department1	Medium	1	0		Edit...
5	Person5		Department1	Medium	1	0		Edit...
6	Person1-2		Department1	Medium	1	0		Edit...
7	Person2-2		Department1	Medium	1	0		Edit...
8	Person3-2		Department1	Medium	1	0		Edit...
9	Person4-2		Department1	Medium	1	0		Edit...
10	Person5-2		Department1	Medium	1	0		Edit...
11	Person6-2		Department2	Medium	1	0		Edit...
12	Person7-2		Department2	Medium	1	0		Edit...
13	Person8-2		Department2	Medium	1	0		Edit...
14	Person9-2		Department2	Medium	1	0		Edit...
15	Person10-2		Department2	Medium	1	0		Edit...
16	Person6		Department2	Medium	1	0		Edit...
17	Person7		Department2	Medium	1	0		Edit...
18	Person8		Department2	Medium	1	0		Edit...
19	Person9		Department2	Medium	1	0		Edit...
20	Person10		Department2	Medium	1	0		Edit...

Figure 5.32 – Departmental staffing: Functional structure

The position staffing is shown in Figure 5.33.

Position 1

	Name	Value	Unit
	Person1	1	#
	Person2	1	#
	Person3	1	#
	Person4	1	#
	Person5	1	#
	Person1-2	1	#
	Person2-2	1	#
	Person3-2	1	#
	Person4-2	1	#
	Person5-2	1	#

Position 2

	Name	Value	Unit
	Person6	1	#
	Person7	1	#
	Person8	1	#
	Person9	1	#
	Person10	1	#
	Person6-2	1	#
	Person7-2	1	#
	Person8-2	1	#
	Person9-2	1	#
	Person10-2	1	#

Figure 5.33 – Positional staffing: Functional structure

After we simulated the above three scenarios we found considerable differences between the functional and the divisional structures. These are summarized in the table below. As a result of the differences, the team decided to build an additional structure where the decision making regarding allocation of resources would be taken out of the purview of the positions in the divisional models.

11. Lower Level Divisional Organization (case 'divisional – lower level')

In this model (Figure 5.34) the departmental and task structures are exactly the same as in Section V.F.8 (i.e., divisional organization). However, four new positions have been defined, two each below positions 1 and 2. These positions have been staffed with five people each. Position Skill A1 is staffed with 5 people of Skill A from Department 1 and is responsible for Task 1. Position Skill B1 is staffed with 5 people from Department 2 (Skill B) and is responsible for Task 2 (Figure 5.35). Positions Skill A2 and Skill B2 are similarly staffed with the remaining people from departments 1 and 2 and are responsible for tasks 3 and 4 respectively.

Defining the positions in this manner should eliminate the coordinated decision making required of the positions as in above models. Whereas, earlier, positions had to decide whom to allocate for each task, that decision has already been made given this structure.

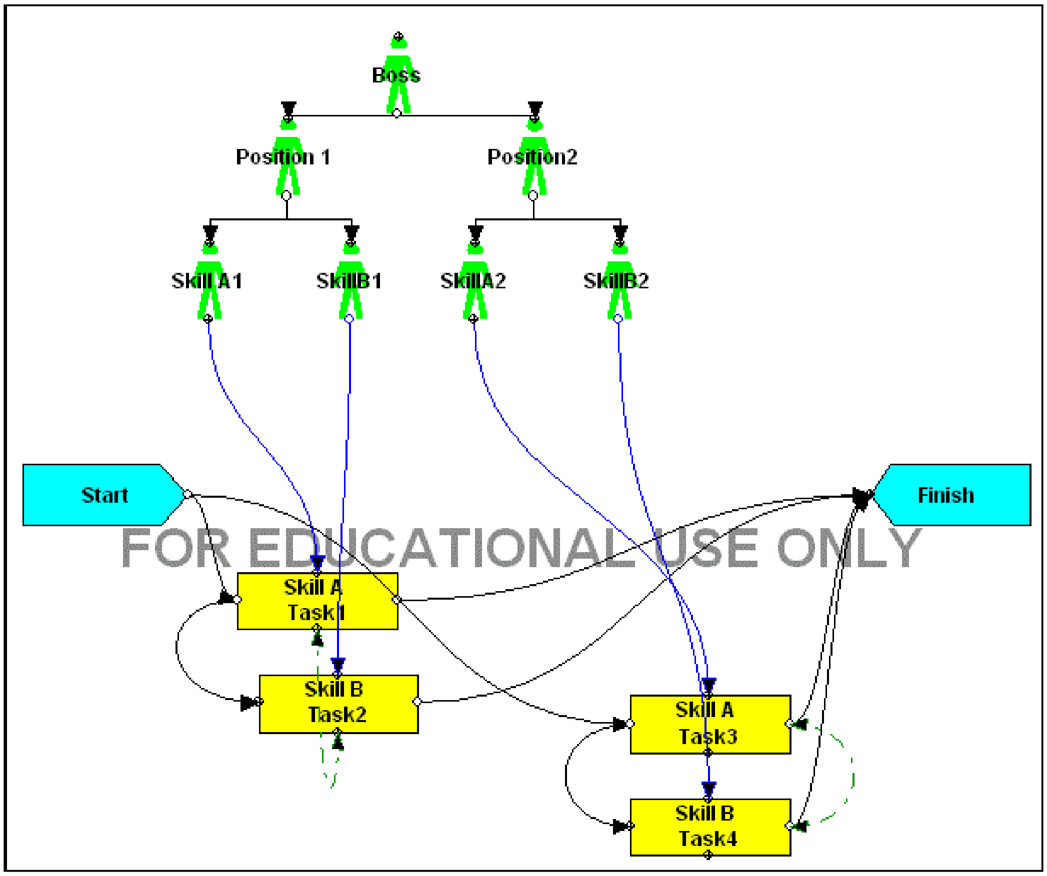


Figure 5.34 – Lower level divisional structure

The positional staffing is shown in Figure 5.35.

Skill A1

Name	Value	Unit
Person1	1	#
Person2	1	#
Person3	1	#
Person4	1	#
Person5	1	#

Skill B1

Name	Value	Unit
Person1-2	1	#
Person2-2	1	#
Person3-2	1	#
Person4-2	1	#
Person5-2	1	#

Skill A2

Name	Value	Unit
Person6	1	#
Person7	1	#
Person8	1	#
Person9	1	#
Person10	1	#

Skill B2

Name	Value
Person6-2	1
Person7-2	1
Person8-2	1
Person9-2	1
Person10-2	1

Figure 5.35 – Positional staffing: Lower level divisional structure

The four positions have been staffed such that each position is responsible for one task and the skills of the position match the skills required to execute the task.

12. Summary and Comparison

The simulated results from the four models are presented below in tabular form:

Case	Divisional	Functional	Divisional Reversed Staffing	Divisional Lower level
Task 1	186.3	140.3	186.3	144.2
Task 2	186.3	140.3	189.0	144.2
Task 3	189.1	141.0	189.0	144.0
Task 4	189.1	141.1	189.1	144.1

Table 5.2 – Task duration comparison of the four scenarios

From the above table it is clear that the functional organization is better at performing the tasks as compared to the two divisional scenarios. There is no appreciable difference in performance between the Divisional and the reversed staffed Divisional structures indicating that the nature of staffing alone does not affect the outcome. Another interesting observation was that the lower level divisional model was comparable in performance to the functional model. This would seem to indicate that the major reason for the poorer performance of divisional scenarios is the manner in which VDT assigns actors to tasks. In both the divisional scenarios, each position had ten people (five of each skill), and the position was expected to assign five people to each task depending upon the skill requirement. Detailed project statistics comparison of the models is shown in Table 5.3.

Case	Divisional	Functional	Divisional Reversed Staffing	Divisional Lower level
Simulated Duration (Days)	135.6	101.4	135.4	104.2
CPM Duration (Days)	100	100	100	100
Total volume	2024.4 days	2013.2 days	2023.5 days	2037.5 days
Work volume	2000.0 days	2000.0 days	2000.0 days	2000.0 days
Rework volume	18.8 days	9.8 days	17.7 days	22.8 days
Coordination volume	5.4 days	3.4 days	5.7 days	8.0 days
Decision wait volume	0.2 days	0.1 days	0.1 days	6.7 days
FRI	0.497	0.503	0.537	0.475
PRI	0	0	0	0
Coordination risk	0.514	0.513	0.509	0.386
Communication risk	0.77	0.77	0.77	0.58
Meeting risk	0	0	0	0

Table 5.3 – Project properties comparison of the four scenarios

The difference in performance between the functional and the lower level divisional model lies mainly in the rework, coordination and decision wait volumes. It seems that the more hierarchic organization (lower level divisional model) results in higher values for all of these thus degrading performances. But the same organization results in improving the coordination and communication risks. The absence of Project Risk Index (PRI) suggests that we were not able to capture the project level coordination and this may explain part of the discrepancy in results. However, due to paucity of time, we could not further experiment with the models and decided to document our findings.

An explanation for these differences between Functional and Divisional lower level structures is not readily apparent and we leave further exploration of these questions to those who do follow up work on this effort.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

This concluding chapter argues that VDT has great potential that can be exploited further to augment DDD experiments. Despite our moderate success in fully replicating many of the DDD/Experiment 8 contextual behaviors, we have shown how these can be studied further using a "building blocks" approach – starting with simple models to isolate the relevant behaviors, and then integrating those within a more comprehensive scenario and more complex organization structures. We describe how this project contributes to our knowledge in computational simulation systems and their possible applications to A2C2 experiments. We also discuss some limitations of this work and natural extensions for future work.

A. SUMMARY OF PROJECT RESULTS

In this study we present evidence that computational simulation systems (i.e., VDT) can be used to emulate experiment results. It is only the paucity of time that prevents us from going further with the models and capturing other relevant DDD variables (e.g. the project level coordination). However, by documenting our findings, we leave the door open for future teams that may want to expand upon this work.

1. Usefulness of Computational Simulation

Because organizational experimentation in the field is costly and time-consuming, and because controlled experiments in the laboratory are difficult to run as provide data based on small sample sizes, organizational simulation is a compelling method for understanding many kinds of real-world behavior.

It can be a powerful tool for generating and filtering testable hypotheses to be further evaluated in field and laboratory experiments. Furthermore, it can provide insight into what experimental data would be important to collect. With all these considerations in mind, and given the observed capability of using VDT to model DDD, our team considers this project worth the effort of undertaking it.

2. Context

Organization theory describes the concept of "organizational congruence" as a factor influencing organization effectiveness. This has been demonstrated by recent results from C2 experiments that have shown that "*the better an organization is matched structurally to the overall mission, the better will that organization perform – and that mismatches are potential drivers for adaptation of organization structure"* (Diedrich et al., 2003).

The results of Experiment 8 (Kleinman et al., 2003), clearly show that performance - broadly defined as the percentage of tasks completed - was significantly higher in the congruent cases (fit between organization and mission) compared with incongruent cases (misfit between organization and mission).

Our primary goal in this project was to determine if and how we could use VDT to emulate the results obtained from A2C2 Experiment 8. This required adapting the VDT tool to the domain of military command and control.

3. Strategies and Results

Starting with a core of DDD contextual micro-behaviors derived from lab observations, we use a building-blocks strategy in our attempt to capture those key behaviors with VDT. In Chapter 5 we describe the 8 selected DDD behaviors, the solutions adopted to model them, and some limitations.

While the team was successful in modeling behaviors such as task precedence, coordination requirements and department and staffing, for others such as geography, expendable assets, the impact of coordination or workload, and task prioritization the results were not as expected. And although we found solutions for two of these initial anomalies (time criticality of tasks and geography), some limitations still applied. In the case of *time criticality of tasks* the proposed solution was to use the 'high priority task switch' available in VDT constraining the actors to expend resources in executing these priority tasks first. However, this solution could not capture the option that actors have in DDD to skip one of these critical tasks; in VDT, once modeled, the task will be executed.

For *geography* the solution adopted was to assign tasks to actors within their geographical regions in accordance with the DDD scenarios, irrespective of their skills. One major limitation still exists here, because this solution does not account for the time period required for an asset (rocket) to reach the assigned target.

After modeling the selected individual behaviors and implementing all the solutions described in section 5.3, we built two structures D-d and F-d and compared the simulated results. Our aim here was to observe the dynamics of a complex model after all the projected solutions in section 5.3 are included. Also, we seek to identify and isolate behaviors associated with VDT that are consistent or in contrast with DDD.

Our results show that coordination requirements and decision wait times are much higher in the divisional organization (congruent with task scenario) than in the functional organization (incongruent with the task scenario). This is exactly opposite to our assumptions that congruent structure/scenario should perform better. Therefore, we proceeded to isolate and understand the behavioral nuances of the two simulations by building simple representative models, developed in an incremental fashion. The results indicate that factors such as communication links, organizational hierarchy and staffing do not have much explanatory power for the difference between scenarios. This finding may have implications for the ultimate success of using VDT to capture design principles used in DDD because the latter relies significantly on coordination as a defining factor of congruence and hierarchy plays a significant role in military organizations. Further exploration of these factors is required.

The last step was to model four simple representative structures (2 positions/four tasks) and to compare the results. While the task scenario was kept the same across all four models, three different versions of a divisional organization and one of functional were built. The first divisional structure (divisional) involved mixed skills and staffing from two departments, the second divisional structure (reverse staffed divisional) used mixed skills but staffing from a single department while the last divisional structure (lower level divisional) was more hierarchical developed, introducing 2 sub-positions below each existing position. In the latter structure (lower level divisional) the task assignment was divided between the sub-positions, for each set of tasks.

The results indicate that there is no appreciable difference in performance between the divisional and the reverse staffed divisional scenario, which leads to the conclusion that the nature of staffing alone does not affect the outcome. Also, as the lower level divisional model is comparable in performance to the functional model, we conclude that the major reason for the poor performance of divisional scenarios could be the manner in which VDT assigns actors to tasks.

B. CONTRIBUTIONS

We have shown that emulating organizational behavior using computational simulation models is a valuable strategy for generating insight into organizational contextual behavior. Our project contributes to a better understanding of the advantages and the possible limitations of such work, showing that the task of building computational models to replicate experiment results is not an easy one but also not impossible. Our work also clearly illustrates that VDT has great potential and that individual behaviors needed in the modeling and design process could be isolated and further examined.

C. WEAKNESSES

Due to lack of time and challenges of modeling critical DDD behaviors with VDT, we cannot further, analyze and compare the results against the observed organizational performance during A2C2 Experiments. As a consequence, we limit our final work to analyzing and interpreting the VDT simulated results and to documenting the gains and limitations of our work. Hopefully this will assist people pursuing this topic in the future and provide more in-depth examination of how VDT can be used as a pre-experimental modeling tool for A2C2 research.

D. CONCLUDING REMARKS

Contextually changing behavior is a common phenomenon in military organizations, and the structural match of an organization to the overall mission leads to better performance.

Also the mismatches are potential drivers for adaptation (Diedrich et al., 2003). Experiments using simulation models have yielded insights into the magnitude and direction of certain contextual effects. Still, much research into contextual behavior remains to be done, both theoretically and empirically, and VDT can be a viable tool to this goal.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Diedrich, F.J, Hocevar, S.P, Entin, E.E., Hutchins, S.G., Rubineau, B., & Mac Millan, J. (2003). “When Do Organizations Need to Change (Part I)? Coping with Incongruence”

Kleinman D.L, Levchuk G.M., Hutchins S.G., and Kemple W.G., “Scenario Design for the Empirical Testing of Organizational Congruence”, 2003

Levit, R. E., VDT Computational Emulation Models of Organizations: State of the Art and Practice, *Overview of the Virtual Design Team (VDT) Research Program*, at <http://www.stanford.edu/group/VDT/VDT.pdf, April 5, 2004

Jin, Yan, and Raymond E. Levitt, "The Virtual Design Team: A Computational Model of Project Organizations,"Journal of Computational and Mathematical Organization Theory 2 (3), Fall, 1996, pp. 171-195

Levitt, Raymond E., “Organizational Analysis and Design Tools: State of the Art,” First International Conference on Computational and Mathematical Organization Theory, Monterrey, Mexico, October 30, 1996. [1996 CP 42]

Nissen, Mark E and Levitt, Raymond E, “Dynamic Models of Knowledge-Flow Dynamics”, CIFE Working Paper #76, November 2002

Kleinman, D.L., Young, P.W., Higgins, G., “The DDD-III: A Tool for Empirical Research in Adaptive Organizations”, 2003

Wong, A., Kleinman, D.L., “DDD Post Processor V1.2 User Guide”, Oct 4, 2002, Naval Postgraduate School, Monterey, CA

Fridsma D., “Organizational simulation of medical work: An information processing approach”, Stanford, CA, (2003), Stanford University

SimVision™ Tutorial: Building Basic models, Revision 5: January 16, 2003, ePM, LLC

SimVision™ Users Guide, Revision 7: January 15, 2003, ePM, LLC

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

2. Professor Susan Hocevar
   Naval Postgraduate School
   Monterey, California

3. Professor David Kleinman
   Naval Postgraduate School
   Monterey, California

4. Professor Mark Nissen
   Naval Postgraduate School
   Monterey, California