

# Use of Dynamic Models and Operational Architecture to Solve Complex Navy Staffing Challenges

Darby Grande\*, J. Todd Black\*\*, Jared Freeman\*, Tim Sorber\*\*, Daniel Serfaty\*

\* *Aptima, Inc., dgrande@aptima.com, jfreeman@aptima.com, serfaty@aptima.com*

\*\**Klett Consulting Group Inc., todd.black@kcg-inc.net, tim.sorber@kcg-inc.net*

**Abstract.** The United States Navy established eight Maritime Operations Centers (MOC) to enhance its command and control of forces at the operational level of warfare. Each MOC is a headquarters manned by qualified joint operational-level staffs, and enabled by globally interoperable C4I systems. To assess and refine MOC staffing, equipment, and schedules, a dynamic software model was developed. The model leverages pre-existing operational process architecture, joint military task lists that define activities and their precedence relations, as well as Navy documents that specify manning and roles per activity. The software model serves as a “computational wind-tunnel” in which to test a MOC on a mission, and to refine its structure, staffing, processes, and schedules. More generally, the model supports resource allocation decisions concerning Doctrine, Organization, Training, Material, Leadership, Personnel and Facilities (DOTMLPF) at MOCs around the world. A rapid prototype effort efficiently produced this software in less than five months, using an integrated process team consisting of MOC military and civilian staff, modeling experts, and software developers. The work reported here was conducted for Commander, United States Fleet Forces Command in Norfolk, Virginia, code N5-OLW (Operational Level of War) that facilitates the identification, consolidation, and prioritization of MOC capabilities requirements, and implementation and delivery of MOC solutions.

## 1. INTRODUCTION

The Navy developed the Maritime Operations Center (MOC) concept to enhance its command and control of forces at the operational level of warfare [1]. To oversee the development of the MOC concept, the Navy gave United States Fleet Forces Command (USFFC) the responsibility to standardize MOC staff functions and processes. This standardization will enable interoperability with the joint community and promote commonality across all Fleet and principal headquarters.

USFFC code N5-OLW (Operational Level of Warfare) used the Department of Defense Architecture Framework (DoDAF) to develop Business Process Models (BPM) for MOC processes. These BPMs, called Operational Views (OV-6c) in DoDAF [2], define MOC processes, their sequence, the organizational elements that execute them, and the products of those work activities. The dynamic modeling work reported in this paper transformed the static DoDAF documents into an executable software model called the MOC Performance Assessment Tool (MOC-PAT). The MOC-PAT is designed to support decisions regarding MOC staffing, such as whether a staffing plan is adequate to execute the many MOC processes required to support a specific mission set at a specified operational tempo. The first application of this tool supports planning and execution of Navy exercises to accredit Fleet MOCs.

This paper outlines how a multi-disciplinary team developed an innovative solution for the Navy

leveraging existing architecture products and software modeling approaches. Section 2 of this paper defines the problem. Section 3, describes the technical development of the initial version of the MOC-PAT. This is followed by a discussion of the data used to exercise the model and a presentation of initial results in Section 4. Finally, in Section 5 presents our conclusions and the directions for our future work.

## 2. PROBLEM DEFINITION

The MOC concept is a recent development in the Navy. In order to ensure MOCs meet mission objectives for Fleet and Combatant commanders while implementing necessary interoperability standards, USFFC tasked Commander Second Fleet to establish a MOC Project Team to explore and document MOC doctrine, organization, training, material, leadership, personnel and facilities. As this effort evolved and the MOC Project Team transferred to USFFC as code N5-OLW, it was evident that a means of linking mission tasking to MOC manning and performance was needed to ensure MOCs are staffed and equipped to mission requirements.

USFFC N5-OLW developed BPMs of over 30 MOC processes, documenting hundreds of activities within each process. These BPMs were created using the DoDAF standard OV-6c format. Typically, these diagrams are developed to support acquisition decisions and reside in a central Navy architecture repository, the Syscom Architecture Development and Integration Environment (SADIE). The MOC-PAT leverages these preexisting BPM documents and uses them

to develop accurate models of the operating MOC.

This is accomplished by linking MOC processes back to Joint Mission Essential Tasks Lists (JMETL), first identifying core missions a MOC staff is required to execute and then relating those mission tasks to the associated JMETL tasks. Manning information based on existing MOC manning documents and role data (developed from surveys and onsite observation) is combined with process activity workload observations (i.e., time to complete activities, or work products required to complete activities) to populate the OV-6c BPM documents in the MOC-PAT. These data are then available to support model runs to analyze MOC staff execution and support accreditation events.

### 3. MODEL DEVELOPMENT

The initial version of the MOC-PAT is designed to enable skilled analysts at USFFC N5-OLW to test the impact of MOC manning estimates on MOC performance at the Numbered Fleets executing Normal & Routine (N&R) Missions. In this section we introduce the model, its assumptions, the dynamics, and the output capability for the users.

#### 3.1 Model Introduction

A mission scenario that the user constructs contains a number of *processes*, each of which is made up of a series of *activities*. While the processes are executed, a fixed schedule of *battle rhythm events (BRE)* occurs. It includes special working group meetings and regular briefs to senior staff. The BRE and the activities produce and consume (that is, require) *information products*, and these well-defined products serve as the linkages between different parts of the organization and their many processes and BRE. For example, a planning activity may produce a plan (a document) that is a required input to an assessment activity to communicate which indicators of progress should be monitored. The MOC organization that will accomplish this mission is made up of multiple *organizational units (OU)*. Each OU has several *billets* (individuals) assigned to it, and each billet is assigned a collection of *roles* he may take on, one at a time throughout the mission. These roles currently serve as proxies for more detailed information about billets' associated knowledge and skills, which we hope to incorporate in future versions of the MOC-PAT.

The work discussed in this paper was conducted for an initial proof-of-concept phase, so a number of simplifying assumptions were necessary. As the work continues, we are re-visiting each of these to refine and enhance the model. We assume:

- Billets are available to work 24 hours each day
- The MOC is operating under Normal and Routine conditions
- Each process begins at scheduled times, according to user-specified cycles
- Each activity cannot begin until its preceding activities (within the process) are concluded
- If an activity is prompted to start at time  $t$  (by the schedule or by the conclusion of its preceding activities), then it must conclude at a deadline created by adding the longest required processing time by any of its roles to this earliest triggered start time
- Information products have a user-specified shelf life, after which their level of completion decays. This is to ensure that we capture the fact that an activity which is unable to update or produce an information product on time will affect the ability of an activity which required the information product to execute completely.

#### 3.2 Model Dynamics

The purpose of this effort is to help the Navy determine whether the MOCs as envisioned and instantiated are meeting the mission support and interoperability goals. This specific, evaluation goal led us to implement our model of the MOC in a simulation, rather than pursue optimization of the many variables – staff size, schedule, process step configuration, communication strategies, etc. With this simulation, the MOC-expert user is able to configure the particular mission he would like the simulation to “play,” and the MOC organization is then evaluated against this mission scenario.

More specifically, the model enables analysts to answer several questions about MOC activities:

- **The Activities:** Do activities get the resources they need? Which processes & activities began with incomplete resources: human, information, time? Which activities could not begin at all?
- **The Organization:** Do we have enough staff in the right roles? Which organizational elements & staff were overloaded? Which were under-loaded?
- **The Information Products:** Are the information products complete and current when they are needed? What information was incomplete or missing when it was needed?

Analysts answer these questions in a process that consists of four stages: (1) Populate a database, (2) Configure the data and the model that processes them, (3) Run a mission simulation, and (4) Analyze the results. The analyst then

typically returns to step (2), to refine the configuration and continue analysis iteratively.

Data entry and configuration is conducted using a component, called Adaptive Modeling Environment, that imports data specifying mission activities (tasks), activity information requirements, activity schedules, human resources (number and roles of staff), and organizational structure. The AME provides users with standard lists and graph representations of these data, through which the user can add, delete, or edit most data objects.

The mission simulation, designed collaboratively by the USFFC N5-OLW and the development team, is a discrete-event simulation engine. It drives the assignment and execution of the processes over the course of a mission. This simulation operates as follows.

For the purpose of the model, let

$d_{ij}$  = the amount of time role  $i$  is required to spend on activity  $j$ . (If the role is not required for the activity, then  $d_{ij} = 0$ );

$$d_{\max} = \max_{i=1 \dots R} (d_{ij});$$

$c_m^t$  = the completeness of information product  $m$  at time  $t$ ;

$v_i$  = calculated completeness percentage attainable for the current execution of activity  $i$ ;

$v_i'$  = completeness percentage attained in the most recent execution of activity  $i$ ;

$\alpha$  = activity repair coefficient, that is, the rate at which deficient information products input to an activity are improved by that activity;

$\beta_i$  = minimum completeness threshold for activity  $i$ ;

$\tau_i$  = minimum execution time for activity  $i$ ;

$\sigma$  = completeness decay rate for activities;

$w_1, w_2, w_3$  = the weights used in calculating  $v_i$  to balance the importance of preceding activity completeness, information product input completeness, and fulfillment of roles required,

$$\text{such that } \sum_{k=1}^3 w_k = 1;$$

$N_p$  = the number of activities in process  $p$ ;

$M$  = the total number of information product types.

Additionally, we employ the following variables:

$$r_{ij} = \begin{cases} 1 & \text{if role } i \text{ is required for activity } j \\ 0 & \text{otherwise} \end{cases}$$

$$x_{kj}^t = \begin{cases} 1 & \text{if billet } k \text{ is assigned to activity } j \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$a_{mj} = \begin{cases} 1 & \text{if info. prod. } m \text{ is an input to activity } j \\ 0 & \text{otherwise} \end{cases}$$

$$l_{ij} = \begin{cases} 1 & \text{if activity } i \text{ directly precedes activity } j \\ 0 & \text{otherwise} \end{cases}$$

Each time an activity within a process is prompted to begin (either by the process schedule, or by the completion of all the preceding activities), the activity's potential completion score is computed to determine whether the activity has available the resources it needs: the required roles among available staff members; recently updated information products required by the activity; and required preceding activities. The score calculated for activity  $i$  at time  $t$  is

$$v_i = \frac{w_1}{N_p} \left( \sum_{h=1}^{N_p} l_{hi} * v_h' \right) + \frac{w_2}{M} \left( \sum_{m=1}^M a_{mi} * c_m^t \right) + \frac{w_3}{R} \left( \sum_{n=1}^R r_{ni} x_{bi}^t \right)$$

The activity begins immediately if  $v_i \geq \beta_i$ , that is, if the score is above the minimum completeness threshold. (The user can define this threshold differently for each activity to reflect varying priorities for the resources). If the score is not sufficient (the activity does not have enough of the required resources available), the activity will delay its start. The required completion deadline for the activity remains fixed, so any delay in the activity start time reduces the overall duration of activity execution. As the duration of the activity is reduced, the overall quality of the actions, communications, and products of an activity declines.

At each time interval after the initial time at which the activity was prompted to begin, the activity's score is recalculated: increased with the possible addition of any newly available resources, and decreased by the decay rate due to the shorter time for execution and any resources that have become unavailable during the delay. That is, the new score is computed after a starting delay,  $\delta$ , as  $v_i(\delta) = v_i - ((1 + \sigma)\delta)$ . If  $v_i(\delta) \geq \beta_i$ , then the activity may begin. Otherwise, the delay is continued until (1) the activity is able to begin, or (2) the delay has lasted too long ( $d_{\max} - \delta < \tau$ ) at which time the activity fails.

The overall quality of activity is measured by the activity's completeness score, which conveys to subsequent activities thus propagating the effects of shortages of input resources and time. The staff of subsequent activities can partially repair the deficiencies of prior activities, and this is represented by a multiplier on incomplete input we call the repair rate, whose effect grows with the

actual duration of the task<sup>1</sup>. This repair rate is employed to calculate the concluded activity's completeness:  $v_i' = \min(\alpha * v_i, 1)$ . The calculations given for activities throughout this section are used similarly to compute completeness percentages for the BRE.

### 3.3 Model Output

The output of the simulation consists of several measures, which are presented graphically within the software tool to help the analyst rapidly diagnose deficiencies in the staffing plan and mission schedule, and to refine their configuration. These measures are:

- **Activity Completeness:** For each activity that is executed in the mission, we calculate its completeness as the weighted sum of the states of its required inputs at the start of the activity, augmented by a 25% repair rate. The Input Weights are configurable by the user for each activity in order to capture variations in requirements across the three input categories: required information products, required roles, and required completion of prior activities.
- **Manning Employment:** As a mission simulation evolves, organizations dedicate staff (billets) in suitable roles (specific knowledge and skill packages) to activities. Each staff member takes on one of potentially many) roles at a time. For each organization element, we return the percent employment (0 – 100%) of its staff over time. For each role, we return over time the percentages of all the billets capable of fulfilling the role that are currently employed in the role. Finally, for each billet, we return the instantaneous and average workload over the course of the mission. Instantaneous workload is currently dichotomous, as the billet is either employed or is idle.
- **Information Product Completeness:** Each information product has a shelf life that is configurable by the user. Each time an information product is updated by an activity or battle rhythm event, its completeness returns to 100% and remains there for the duration of the shelf life. After this time, the completeness of the information product decays as the information becomes increasingly outdated. For each information product, we return its completeness measure

each time that it was required as input by an activity or a battle rhythm event.

The output of the model has thus far proved accurate and useful when compared to actual MOC staff process execution as observed by USFFC N5-OLW Subject Matter Experts, and during an initial application to a MOC accreditation exercise, as discussed below.

## 4. ACCREDITATION DATA AND RESULTS

In 2008, the Chief of Naval Operations mandated that each MOC be accredited to validate its proficiency at MOC core tasks. The MOC-PAT is used to support this process by analyzing the performance of selected MOCs during accreditation. MOC accreditation is accomplished by USFFC via on-site observation of the MOC staff during a “stressing” event such as a major military exercise. These exercises can span weeks and involve hundreds of MOC staff members exercising a complex combination of the processes based on an assigned mission. The accreditation team must place its few observers where and when stress is likely to show its effects, and conduct analyses that help the MOC refine its staffing, schedule, and processes. In the section below, we discuss the types and sources of the data used for the initial MOC-PAT demonstration and evaluation, and present our initial findings based on these data.

### 4.1 MOC Data Types and Sources

Because the emphasis of this work was to develop a model that could be in use by the end of its initial six-month development period, populating the model with operationally-relevant data was of vital importance. The data required to run the model are billet information, which can be imported from existing command manning documents or manually input through the MOC-PAT configuration interface; role information, which specify the jobs or roles needed to accomplish activities (note that multiple roles can be assigned to individual billets); process diagrams imported from the approved OV-6c diagrams; the “Battle Rhythm”, or daily schedule of leadership meetings and roles of attendees; and the information products that each activity in the process model requires and creates. The data used in the MOC-PAT originate from authoritative sources: billet information from Activity Manning Documents; role information from on-site observation, as well as survey results and workshop interviews; process diagrams from the SADIE architecture repository; and the Battle Rhythm from the MOC's schedule. Additionally, an analysis of the assigned mission is conducted, and mission specific tasks from the Universal Joint Task List (UJTL) are identified. The analyst selects these mission tasks in the software

---

<sup>1</sup> By design, none of the algorithms implemented in MOC-PAT are stochastic in nature at this time; that is, none injects variance into the dataset.

configuration editor, and the MOC PAT then automatically identifies the processes to run based on a mapping by USFFC N5-OLW of tasks to processes. After this initial data import and input, the analyst can generate additional configurations easily in the model, and specify the length of a given mission to test the durability and reliability of an organizational configuration.

The software typically runs each modeled mission simulation in less than a minute, allowing users to rapidly assess and reconfigure the organization as required. Each simulation run produces graphs illustrating workload on staff, process execution success, and the availability of information products to subsequent processes during the simulation. Analysts use these outputs to assess effectiveness of an organizational configuration, to diagnose potential failures, and to specify solutions.

#### **4.2 MOC-PAT Initial Outcomes**

The MOC-PAT was tested during a major Fleet exercise in the spring of 2009. Initial testing indicated that the MOC-PAT results are consistent with observed outcomes in the MOC when reliable data are used and processes in the model adjusted to reflect how the MOC staff conducts its mission tasking.

During the spring 2009 exercise, the MOC-PAT identified several areas of interest that were not noted during on-site observation. These findings were discovered during the exercise, because reconfiguring and running the MOC-PAT was so rapid. The findings helped focus the efforts and attention of on-site observers, and allowed identification of how the MOC staff had spontaneously developed workarounds for some issues. These discoveries were documented as "best practices" to share with other MOC staffs. Observers confirmed other problem areas identified in model runs during on-site observation. These discoveries provided confidence that the model was accurately describing how a MOC staff coped with an assigned mission set. The MOC-PAT was also used to explore how process synchronization and staffing issues might evolve over time, by running the model for missions sets that were far longer than those executed in the live exercise. This analysis identified issues for the MOC staff to explore after the exercise was complete.

#### **5. CONCLUSIONS AND FUTURE WORK**

This first iteration of the MOC-PAT proved the value of executing an operational architecture in software to assess complex Navy organizations and their processes. The MOC-PAT accurately modeled an operational staff's performance, and can provide analysts with insights into issues of

staffing and scheduling of complex process flows. The speed of configuration and simulation enabled analysts to rapidly revise the model to diagnose performance failures and test alternative configurations of the organization.

The next iteration of the MOC-PAT will include more advanced analysis tools, including reports that will support analysis and reporting by a MOC assessment team. In addition, the model is being revised to show the impact of role experience and proficiency on process execution speed (e.g., inexperienced personnel in a billet should slow activity execution while experienced staff accelerate activities.) The next version will also model shifts with greater fidelity than the current version.

In the Fall of 2009, the MOC-PAT will be used to support accreditation team observation of a Fleet MOC staff. The model is also intended to support MOC manning levels determination using data from a separate effort to identify MOC staff competencies and activity durations.

The MOC-PAT makes innovative use of an operational architecture (DoDAF OV-6) by providing a configurable, scalable, valid and executable representation of Fleet MOCs. This fusion of authoritative architectural data with simulation technology has proven to be a cost effective way to analyze complex organizational structures and human interactions.

#### **REFERENCES**

- 1, Department of the Navy (2009), *MOC Fact Sheet (Updated April 30, 2009)*. [https://www.portal.navy.mil/mhq-moc/fltcomm/Shared%20Documents/Fact%20Sheets/MOC%20Fact%20Sheet\\_043009.pdf](https://www.portal.navy.mil/mhq-moc/fltcomm/Shared%20Documents/Fact%20Sheets/MOC%20Fact%20Sheet_043009.pdf).
2. DoD (2007) *DoD Architecture Framework Version 1.5*. 23 April 2007. [http://jitc.fhu.disa.mil/jitc\\_dri/pdfs/dodaf\\_v1v1.pdf](http://jitc.fhu.disa.mil/jitc_dri/pdfs/dodaf_v1v1.pdf).