informs
OR

informs
http://pubsonline.informs.org/journal/inte/

INTERFACES
Articles in Advance, pp. 1–17
ISSN 0092-2102 (print), ISSN 1526-551X (online)

Improving U.S. Navy Campaign Analyses with Big Data

Brian L. Morgan,[a] Harrison C. Schramm,[b] Jerry R. Smith, Jr.,[c] Thomas W. Lucas,[d] Mary L. McDonald,[d] Paul J. Sánchez,[d] Susan M. Sanchez,[d] Stephen C. Upton[d]

[a] Operations Research Department, Naval Postgraduate School, Monterey, California 93943; [b] CANA Advisors, Pacific Grove, California 93950; [c] Naval Surface Warfare Center, Bethesda, Maryland 20817; [d] SEED Center for Data Farming, Operations Research Department, Naval Postgraduate School, Monterey, California 93943

Contact: blmorgan@nps.edu (BLM); harrison.schramm@gmail.com (HCS); jerry.r.smith1@navy.mil (JRS); twlucas@nps.edu (TWL); mlmcdona@nps.edu (MLM); pjsanche@nps.edu (PJS); ssanchez@nps.edu (SMS); scupton@nps.edu (SCU)

Received: February 15, 2016
Revised: October 31, 2016
Accepted: January 27, 2017
Published Online in Articles in Advance: September 6, 2017

https://doi.org/10.1287/inte.2017.0900

Abstract. Decisions and investments made today determine the assets and capabilities of the U.S. Navy for decades to come. The nation has many options about how best to equip, organize, supply, maintain, train, and employ our naval forces. These decisions involve large sums of money and impact our national security. Navy leadership uses simulation-based campaign analysis to measure risk for these investment options. Campaign simulations, such as the Synthetic Theater Operations Research Model (STORM), are complex models that generate enormous amounts of data. Finding causal threads and consistent trends within campaign analysis is inherently a big data problem. We outline the business and technical approach used to quantify the various investment risks for senior decision makers. Specifically, we present the managerial approach and controls used to generate studies that withstand scrutiny and maintain a strict study timeline. We then describe STORMMiner, a suite of automated postprocessing tools developed to support campaign analysis, and provide illustrative results from a notional STORM training scenario. This new approach has yielded tangible benefits. It substantially reduces the time and cost of campaign analysis studies, reveals insights that were previously difficult for analysts to detect, and improves the testing and vetting of the study. Consequently, the resulting risk assessment and recommendations are more useful to leadership. The managerial approach has also improved cooperation and coordination between the Navy and its analytic partners.

History: This paper has been accepted for the special issue on Applications of Analytics and Operations Research in Big Data Analysis.

Funding: The SEED Center portion of this work was funded by the Naval Postgraduate School's Naval Research Program [Projects P14-0487 and NPS15-0021].

Keywords: enterprise risk assessment • project management • data farming • defense • simulation

Like many organizations, the U.S. Navy relies on a team of operations research (OR) professionals, both in house and external, to provide assessments of options under uncertainty. The Assessment Division, Navy Headquarters Staff (OPNAV N81) leads enterprise risk assessments for the Navy. The centerpiece of these assessments is a holistic, end-to-end, scenario-based study, known as campaign analysis (Hughes 1999, Kline et al. 2011). Campaign analysis is a broad-based team effort that requires close cooperation and coordination with other military services and groups to construct scenarios that can be assessed using simulation. Currently, these analyses are conducted in a large-scale stochastic simulation; we and many of our partners use the Synthetic Theater Operations Research Model (STORM) (Group W 2012a, b, c). Originally developed for the U.S. Air Force, STORM now includes a wide range of land, maritime, amphibious, air, space, and logistical facets of modern warfare.

Campaign model construction can be very costly in terms of both input data and developer time. The output analyses can also be extremely demanding and time consuming because of the vast amount of complex output data generated. In this sense, campaign analysis is inherently a big data problem. Campaign simulations are complex models, involving as many as 100,000 input factors (Saeger and Hinch 2001), many of them estimated or uncertain. STORM is no exception. Furthermore, campaign analysis studies are conducted cooperatively with other military services and organizations to accommodate the capabilities, concerns, and equities of the entire Joint force. Such analyses frequently inform legally mandated senior (Cabinet-level) decisions, and have little or no tolerance for schedule slippage. Therefore, it is necessary to minimize risk—both managerial and technical. The decisions based on risk assessments performed using STORM are important (Barber 2014, Beall 2015) and form the basis for defense budget decisions (United States Office of Management and Budget 2016). While the value of a nation's security cannot be quantified, it is noteworthy that the modeling and analysis suite and techniques we discuss in this paper form the analytic underpinnings of the Department of the Navy's annual budget, which is currently over $160 billion.

Literature Review

Supporting enterprise-level decisions under uncertainty, also known as enterprise risk assessment, is a classic OR endeavor. The application of mathematics to military problems has a rich history, dating as far back as Sun Tzu's writings in 500 BCE (Griffith 1963). Modern application of mathematics to military problems at the campaign level has been strongly influenced by F. W. Lanchester's seminal paper (Lanchester 1916). Natural extensions of Lanchester's ideas are regularly incorporated into campaign models used around the world. Military applications were further developed by Morse and Kimball (1951), who coined the term "operations research" to describe their work. Additional references for military OR include Bracken et al. (1995), Wagner et al. (1999), Loerch and Rainey (2007), and Washburn and Kress (2009). Early applications of campaign analysis focused on deterministic methods. Subsequently, stochastic models amenable to closed-form solutions were developed for simplified forms of both duels and full battles; see Kress and Talmor (1999) for examples.

Prior to the 1950s, campaign models were limited to those with analytical solutions. With the advent of digital computing in the mid-20th century, the models rapidly grew in scope, complexity, and realism (Lucas et al. 2015). Indeed, we now have a trillion-fold more processing power than those pioneer military OR analysts had half a century ago. Nevertheless, modern campaign simulation models like STORM can take hours or days to evaluate using today's fastest computers—and the computational analysis may take as long or longer than the model runtime, due to the volume and complexity of the output.

Campaign Analysis in the Modern Era

A warfighting campaign is an example of a complex system with big data challenges. Campaigns can span months, involve tens of ships and battalions and hundreds of aircraft and installations, all executing thousands of separate missions encompassing tens of thousands of exchanges of fire and other engagements. To contextualize the discussion, consider the historic campaign of Guadalcanal (1942–1943). That seven-month campaign included a series of different battles (e.g., amphibious, air, surface, subsurface, and land) and missions (e.g., attack, bombardment, resupply, interdict, construct, and reconnaissance). Each battle and mission created causal threads that coalesced into the final outcome. Similarly, in a simulated campaign, the various entities request hundreds of thousands of logistics resources (e.g., fuel and weapons), make millions of sensor observations (e.g., detection and identification), and receive tens of millions of orders. Every simulated day involves multiple decision cycles, each of which includes tens of millions of considerations yielding hundreds of millions of actionable options.

The complexity of a campaign grows exponentially with the number of entities—such as units, facilities, sensors, or weapons. For campaigns modeled in STORM, every run generates outputs that log the events associated with all entities and their interactions. Thus, a single STORM run typically generates many gigabytes (GBs) of densely packed activity data that must be transformed to produce useful insights. The analyst's challenge is to detect and trace relationships within each run, or trends across the set of stochastic runs, to seek to understand how specific investment option(s) (e.g., ship, aircraft, installation, sensor, and weapon) affect the simulated campaign and contribute to the associated risk assessment.

Another challenging aspect of campaign analysis is assessing the validity of model outcomes. Enormous detail goes into setting up and vetting a STORM scenario. However, because STORM is a campaign analysis simulation that models future possibilities, it belongs to a class of models for which it is impossible to quantitatively compare its final results directly against ground truth (Hodges 1991). Invariably, these models and the credibility of their findings are judged largely by human subject matter experts. This puts a premium on the transparency of both the scenario-building process and the model runs.

Managerial Approach and Controls

Before the analysis team can tackle the big data challenges of the campaign's outcome, it first instantiates a campaign scenario within the STORM simulation environment. That process has its own set of issues and big data analytical needs. The Navy's managerial approach to crafting and instantiating a campaign analysis via a complex simulation project is critical to achieving our twofold study objectives: (1) creating insights and recommendations that can withstand the scrutiny of many stakeholders, and (2) maintaining a study timeline that supports the Navy's annual budgetary decision cycle.

Because so much is at stake, the campaign simulations must be acceptable to many diverse parties. This requires that entities behave and interact in credible ways, both individually and collectively. The degree and nature of these behaviors and interactions are informed by detailed, system-on-system performance data derived from a loosely integrated family of hierarchical higher-fidelity models, experimental tests, and live exercises. All told, N81 uses as many as 20 models for these purposes.

In addition to obtaining stakeholder consensus on the physical aspects included in the model, such as object ranges, speeds, capabilities, and appropriate entity-level behavior, challenges arise involving determining and modeling a suitable concept of operations (CONOPs) and tactical situations (TACSITs) for allied and opposing forces. Each simulated command entity within the campaign must regularly update its perception, assess the state of play, analyze its options, and assign specific tasks to other entities in space and time. The various considerations, which generally evolve over time, are encapsulated in dynamic rule sets. These decisional rule sets must exhibit responses that adequately mimic the preferences, concerns, objectives, risk assessments, and decisions that actual commanders would make under similar circumstances.

In seeking performance realism, the analysts identify, collect, and compile information from across a wide variety of sources (Figure 1). They then interpret and transform that information into performance data and assumptions that are provided to the modelers for integration into the campaign simulation. Similarly, operators and subject matter experts identify and vet the types of activities, concerns, and decisions they expect to occur and a canonical timeline. The decisions are implemented by the modelers as rule sets.

During the scenario development process, STORM trajectories frequently move into unanticipated states. These instances require analysts to quickly ascertain the cause. Was the outcome due to intrinsic uncertainty in stochastic model aspects, poor performance data, data entry errors, insufficient operational assumptions, a limited decision space in the rule set, or some combination of these circumstances? This debugging of the scenario occurs throughout the study's evolution; it involves analyzing the same types and quantities of data as the final scenario.

Organizing for Scrutiny

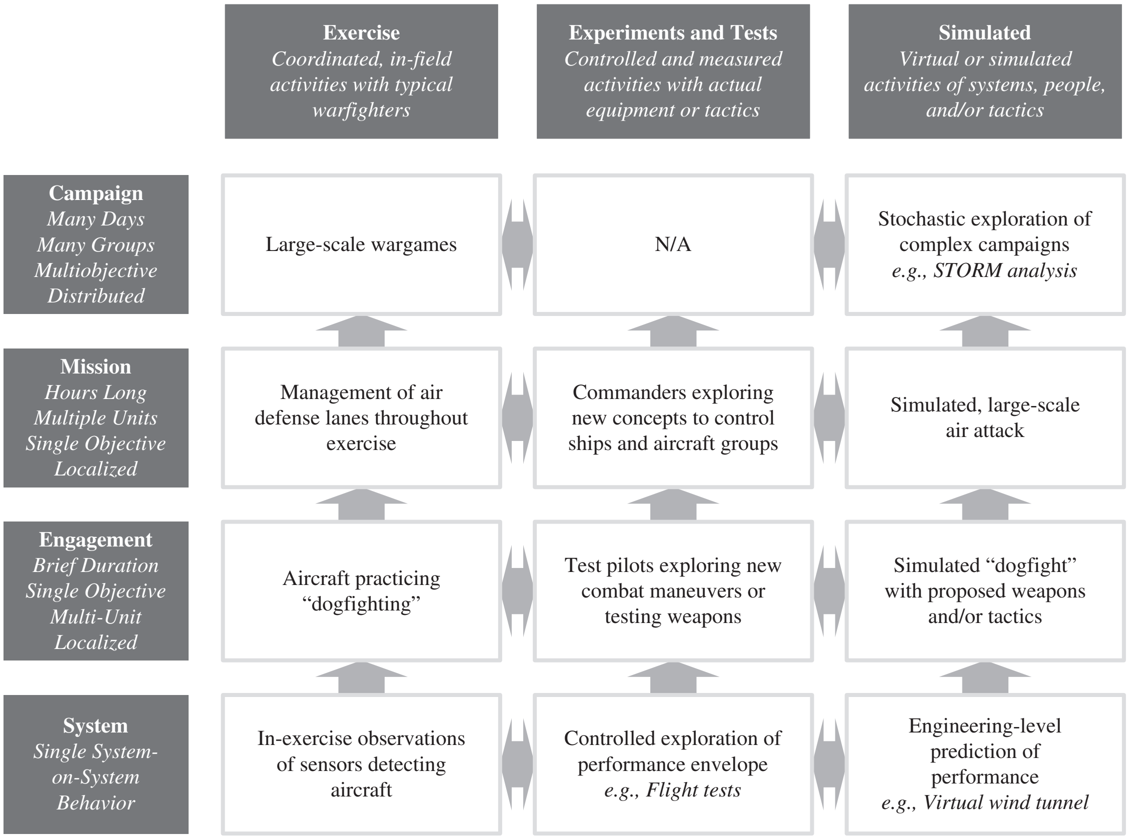
Three distinct technical groups, each with specific roles, emerged from our assessment of the managerial controls: analysts, operators, and modelers. We organize each group beneath an area chief. These three chiefs are peers who must coordinate and communicate with one another in the planning, design, development, review, and analysis of the campaign study. A separate, executive-level project manager, who liaises with external organizations, sets final deadlines, and ensures the overall quality of the effort, oversees the trio. Prior to this reorganization, the Navy vested the responsibilities of all three chiefs in a single person. Although doing so had some advantages in the past, we determined that it had become too much of a burden upon an individual.

To generate a STORM simulation that is capable of withstanding close examination from diverse stakeholders, each team must provide timely and meaningful feedback to the others as the study evolves. We accomplish this in a somewhat unorthodox manner—we treat the entire study as a spiral software-development effort. A benefit of this approach is that, through all the testing and vetting, analysts continuously develop insights and gain intuition into the scenario.

Maintaining a Study Timeline

Treating the study as a software-development effort enhances the defensibility of the analysis and also

Figure 1. Analysts Use a Hierarchy of Sources for Performance Data



	Exercise *Coordinated, in-field activities with typical warfighters*	**Experiments and Tests** *Controlled and measured activities with actual equipment or tactics*	**Simulated** *Virtual or simulated activities of systems, people, and/or tactics*
Campaign *Many Days* *Many Groups* *Multiobjective* *Distributed*	Large-scale wargames	N/A	Stochastic exploration of complex campaigns *e.g., STORM analysis*
Mission *Hours Long* *Multiple Units* *Single Objective* *Localized*	Management of air defense lanes throughout exercise	Commanders exploring new concepts to control ships and aircraft groups	Simulated, large-scale air attack
Engagement *Brief Duration* *Single Objective* *Multi-Unit* *Localized*	Aircraft practicing “dogfighting”	Test pilots exploring new combat maneuvers or testing weapons	Simulated “dogfight” with proposed weapons and/or tactics
System *Single System-on-System Behavior*	In-exercise observations of sensors detecting aircraft	Controlled exploration of performance envelope *e.g., Flight tests*	Engineering-level prediction of performance *e.g., Virtual wind tunnel*

Notes. The rows show the hierarchy of system, engagement, mission, and campaign levels; analysts integrate up the granularity levels to identify how information should be abstracted. The columns indicate the means of data generation, i.e., from operational exercises, experiments, and modeling and simulation; analysts seek consistency across the information sources.

yields another important benefit: maintaining a disciplined study timeline. This occurs for three major reasons.

First, by leading with design, analysts decide up front what analytic insights they need from the study. This enables requirements to be defined early and the STORM simulation to be planned and organized before development fully begins. It reduces the study’s execution risk by identifying gaps in the data; allows the modelers and analysts to negotiate abstractions early on; gives the modelers a clear set of objectives; and sets expectations for the project manager.

Second, using a spiral development process (Boehm 2000) with scheduled releases and expected capabilities, the maturation of a STORM scenario occurs in a controlled manner. We find this incremental approach far superior to our prior single-final-deadline method, which tended to push development and vetting closer to the final briefing to leadership. The enforcement of the new interim deadlines is popular within the team. The modelers appreciate the clearly defined deadlines of the spirals. The analysts and operators benefit from access to evolving testable releases. Leadership now has the ability to qualitatively observe progress, identify concerns, and manage execution risk throughout the project.

Third, testable releases allow the evolving scenario to be reviewed, tested, and analyzed for compliance with the design and consistency with expected behavior. Through the testing process, we can identify problems within the simulation, embedded data, and operational behavior early enough in the development cycle to implement repairs well before we perform the final analysis. These releases also help the analysts to continually refine their knowledge and intuition about the scenario.

All these aspects—spiral development, team organization, development process, and testing procedures—are codified into a project plan that governs the entire study. This plan facilitates the transformation of insights garnered from STORM experiments into actionable recommendations.

Big Data Approaches to Model Analysis

The Setting

Building a full-scale STORM scenario is a massive undertaking. Analysts are required to specify the attributes, capabilities, and decision rules for thousands of entities. Choices are made on how to adjudicate a large number of interactions among entities—and between entities and the environment. The elements that compose the environment in which the campaign simulation will occur all need to be entered, including terrain, roads, weather, elevation, and more. All told, STORM models typically require upwards of 40 megabytes (MB) of data spread over nearly 150 input files. These files contain many types of data in a variety of formats with a myriad of details about individuals, groups, classes of entities, and decision rules that capture CONOPs and tactics. Acquiring, vetting, and verifying the data are challenging tasks involving numerous organizations spread throughout the Department of Defense.

Once the required inputs for STORM to simulate a campaign are ready, multiple replications are run for analyst-selected input combinations using state-of-the-art computing platforms. This is vital for any stochastic simulation. For a large scenario, a single replication can take hours to complete and generates tens or hundreds of GBs of output data. The output contains details about individual entity movements, communications, sensor detections, weapon engagements, and decisions, such as the scheduling of aircraft missions. Some of the data are stored as fields in a large database, and some are reported in large text files full of the results of custom PRINT statements inserted into the code by the modeler. Making sense of this magnitude of data, with its variety of data types and formats, is a classic big data problem. The challenge for analysts is to efficiently glean insights from this massive data—and to transform those insights into actionable recommendations within a tight timeline.

Drawing credible and defensible insights from the above-described mass of data is complicated by the nature of military campaigns. Traditional summary statistics, point estimates, and regression models can readily be compiled even from data sets this large, and are often useful. However, innovative techniques are required to develop a deep understanding of the drivers of the simulation by revealing temporal and spatial patterns and semi-causal relationships—such as when the response may occur probabilistically at a random time after the predictor event.

Data Farming vs. Data Mining

To address these issues, data farming experts were brought into the partnership with OPNAV N81. The goal was to develop new capabilities that enable researchers to more comprehensively analyze a set of STORM replications than was previously feasible while simultaneously reducing the time required. Lucas et al. (2015, p. 297) describe the differences between data mining and data farming with the following metaphors:

> Miners seek valuable nuggets of ore buried in the earth, but have no control over what is out there or how hard it is to extract the nuggets from their surroundings. As they take samples from the earth they gather more information about the underlying geology. Similarly, data miners seek to uncover valuable nuggets of information buried within massive amounts of data. Data-mining techniques use statistical and graphical measures to try to identify interesting correlations or clusters in the data set.
>
> Farmers cultivate the land to maximize their yield. They manipulate the environment to their advantage using irrigation, pest control, crop rotation, fertilizer, and more. Small-scale designed experiments let them determine whether these treatments are effective. Similarly, data farmers manipulate simulation models to their advantage, using large-scale designed experimentation to grow data from their models in a manner that easily lets them extract useful information .... [The output data sets] also contain better data, in the sense that the results can reveal root cause-and-effect relationships between the model input factors and the model responses, in addition to rich graphical and statistical views of these relationships.

Computer-generated big data differs from observational big data in several ways. Although not the focus of this paper, designed experiments can be used to explore structured variations in input factors. This yields data sets that are much bigger than the already-large output data sets from a single scenario, but are still far smaller than the data sets that would be needed to gain insights if the factor combinations were chosen at random or in an ad hoc manner. Massive sensitivity analysis, via large-scale designed experiments, can be quite useful for verification (model debugging) and validation (assessing and improving the model's credibility) during the scenario-building process. See Sanchez (2015) for a more detailed discussion of big data aspects of data farming experiments, Sanchez et al. (2012) for examples of large-scale military simulation experiments, McDonald et al. (2014) for an overview of the initial STORM postprocessing efforts, and Bickel (2014) for a proof-of-concept designed experiment involving STORM.

Data farming and data mining both benefit from cluster computing. Data farming is particularly well-suited to parallelization. Because of its run-oriented nature, it is easy to distribute runs for different design points and replications across multiple processors—either on a single machine or, more commonly, on computing clusters—using readily available software, such as HTCondor (https://research.cs.wisc.edu/htcondor/). Pertinent information can be extracted from the simulation output files on the processing nodes to reduce the data-transfer requirements.

STORMMiner: A New Campaign Analysis Tool

A key enabler of data farming is the ability to automate the process of running the simulation and gathering results into a form suitable for analysis. Accordingly, we created a purpose-built shell running outside STORM, which we called STORMMiner (SEED Center for Data Farming 2015), to tackle the big data challenges discussed above, while increasing both the speed and utility of analytic insight. The objective of STORMMiner is to quickly harvest information from a set of STORM replications and present the user with a suite of new analysis products (e.g., special-purpose tables and graphics) to augment those internal to STORM. These new methods help illuminate previously unexplored areas of the analytic space as well as facilitate quality control and debugging.

STORMMiner is composed of routines built in a combination of Scala (http://www.scala-lang.org) and R (https://www.r-project.org). Scala is a scalable, hybrid object-oriented functional programming language that runs on a Java Virtual Machine (JVM) (http://java.oracle.com) and eases the burden of extracting and manipulating data from very large files. R is a widely used, open-source statistical computing language, which has broad user-community support. STORMMiner can be instantiated on a desktop or laptop running the Linux operating system. It typically has a runtime of one to three hours, based on the number of key metrics selected for analysis, and generates approximately 50 GB of output.

Some of the new capabilities that STORMMiner provides include the following:

- Dynamic sample-size requirement determination with early-termination option;
- A quick-look dashboard;
- Time-series plots, histograms, killer-victim scoreboards, and summary statistics and indications of outliers for losses and key metrics;
- Unit and event execution graphs;
- Cluster analysis to highlight common characteristics shared by bifurcated results (if present);
- Campaign progression and event heatmaps that indicate the status of resources and campaign objectives over time and the location of casualty occurrences; and
- Classification and regression trees that identify patterns in key outcomes as a function of scenario inputs and events.

Illustrating STORMMiner to Explore the Punic Scenario

We illustrate some of the new analysis capabilities on the notional, unclassified Punic scenario, which the STORM developers use to train STORM users. This scenario is a modern variant in a setting loosely based on the wars between Rome and Carthage from approximately 250 to 150 BCE. It includes examples of many of the types of interactions we are interested in studying in an abstract campaign environment—such as naval, land, and air warfare—over an extended span of space and time. The defense community commonly refers to friendly and hostile forces as Blue and Red, respectively. In the updated Punic scenario, Blue represents Carthage and its allies, while Red represents the Swiss Empire (a notional variant of the earlier Roman Empire).

As previously discussed, with computer-generated big data, we determine what and how much data to produce. Thus, a critical choice is to determine how many replications are desired. Replications incur a cost in time and money. In the past, a predetermined sample of 30 was the most common choice of Navy analysts—because this is frequently recommended as enabling the use of the central limit theorem for estimating means. The number of replications required depends on how precisely one desires to be able to estimate key output measures (e.g., losses to a critical system, such as Blue aircraft carriers). To confidently detect large effects relative to the natural variability of the scenario, perhaps far fewer than 30 runs may be needed. Conversely, when effects are small relative to the variance, then substantially more than 30 replications may be required to achieve the desired precision and confidence levels. The precision of the estimates depends on the sample size and the distribution of the output measure. As with many big data applications, the distribution of the output measure (e.g., its shape and standard deviation) is likely to be unknown in advance of running the experiments.

STORMMiner allows analysts to dynamically calculate the sample size required to achieve a desired level of precision, quantified as the half-length of a confidence interval for a specified coverage probability, using Singham's (2014) newly developed sequential stopping rule. Singham's approach uses resampling and is insensitive to the distribution of the simulation output (i.e., it does not need to be normally distributed) and provides coverage probabilities close to those specified. Seymour (2014) contains a detailed review of this algorithm and its performance for multiple metrics from the Punic scenario.

Insights from Big Data Graphics

The ultimate goal of this research was to provide analysts with tools that help them to quickly understand the drivers and relationships within the simulated campaign. From this deep understanding, robust insights will be obtained that can be presented to Navy leadership (and beyond) on what it takes to win (WITTW) the campaign. The following are examples, adapted from the STORMMiner user's manual, of some new software and methods that help analysts identify those factors (e.g., systems, events, interactions, thresholds, and CONOPs) that enable Blue to win the campaign with minimal losses. Of course, these are in addition to classic simulation analysis techniques, such as summary statistics and fitted regression models, which we do not discuss in this paper in the interest of brevity.

An important first step in any large-scale data analysis is to preprocess the data. In this case, based on guidance from experienced STORM developers and analysts, in addition to the Punic input data files, we assembled a summary of key simulation activities and placed it into what we call the metrics table. Each row in the table corresponds to a stochastic replication and each column corresponds to a particular metric or measurement, with the names (i.e., column headings) specific to the scenario.

To provide analysts with a quick summary across a breadth of measures of interest, the metrics table includes the following:

- Loss and casualty data, by side and asset type;
- Status of critical logistics resources;
- Missions flown and cancelled, by aircraft and mission type;
- Binary indicators created by the modeler to capture the status of user-specified activities or events of major importance; and
- Execution data that signal whether a major phase of a unit's command and control plan (C2 plan) has been executed and, if so, at what time.

Determining which information to place into the metrics table, and how to summarize or discretize it, is akin to the feature-extraction phase of a data-mining problem (Witten et al. 2011). These metrics can illuminate previously unexplored areas of the analytic space, and have proven valuable in assisting analysts with quality control and debugging activities.

Given the vast number of entities in the simulation, an important part of preprocessing and postprocessing is the use of automated naming conventions to describe events and conditions in the simulation output data. These can yield long names that may initially appear unwieldy, but are very informative to the analyst. "RedAirSupremacy" is a relatively short label that characterizes whether the Red forces have achieved air supremacy at a particular point in time. "BlueAdvSAMSitesDeadInSouthSpainAOI" is a much longer label that represents the condition that the Blue force advanced surface-to-air-missile sites have been destroyed in the South Spain area of interest. STORMMiner also tries to infer the type of metric (i.e., binary, categorical, or numerical) from the input. Scripts that handle the character parsing and concatenation, after checking to see whether the conditions hold, allow the tools and graphs developed for the Punic training scenario to be readily customized for other entities and conditions in other campaigns.

Visual Summary Tools

The new quick-look dashboard (Figure 2) informs the user how often objectives are met in each instance and is the starting point for exploring the response space. It displays scores of output measures across dozens of runs at a glance. Each row describes a campaign objective specified by the user. Hyperlinks allow researchers to dynamically access other analytic artifacts described below. In this example, Blue carrier losses = 0 is defined as success, whereas Blue carrier losses ≥ 1 is defined as failure. Each cell contains the number of occurrences of the condition for that replication. The green or red color indicates if the threshold condition was met (green) or not met (red).

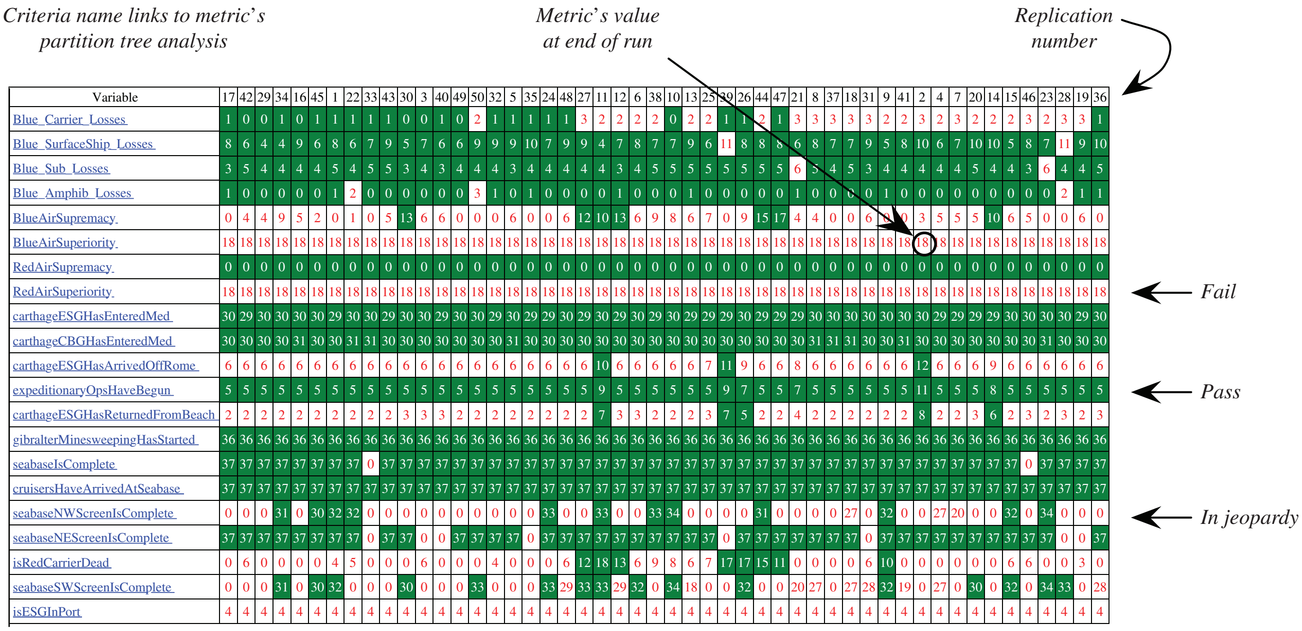
A similar outlier dashboard (not shown) presents analysts with a color-coded map that identifies runs in which discordant data occur for user-specified outcomes.

It is also informative to see how the key metrics relate to each other. The correlation plot of key metrics (Figure 3) can be used to examine a user-specified subset of the metrics. This figure shows two very strong positive correlations and one very strong negative correlation. A few potential insights into this notional scenario follow. For example, the positive correlation between the number of Blue aircraft lost and the number of Red advanced surface-to-air missile (SAM) sites destroyed suggests that destroying Red SAM sites, an important objective, comes at a cost. Additionally, there is a negative correlation between the number of amphibious ship losses and the amount of time Blue has air supremacy. One possibility is that the longer Blue has air supremacy, the more protection the amphibious ships receive. We also observe that the length of time Blue is able to achieve and hold air supremacy is positively correlated with Red carrier losses. These last two correlations are consistent with the conventional wisdom about the importance of achieving early air supremacy.

Condition, Event, and Resource Heatmaps

What, when, and where certain events and conditions took place is critical to understanding a simulated campaign. To assist with these processes, we developed

Figure 2. The Quick-Look Dashboard Shows, in Aggregate, How Often the User-Defined Success Metrics Are Met

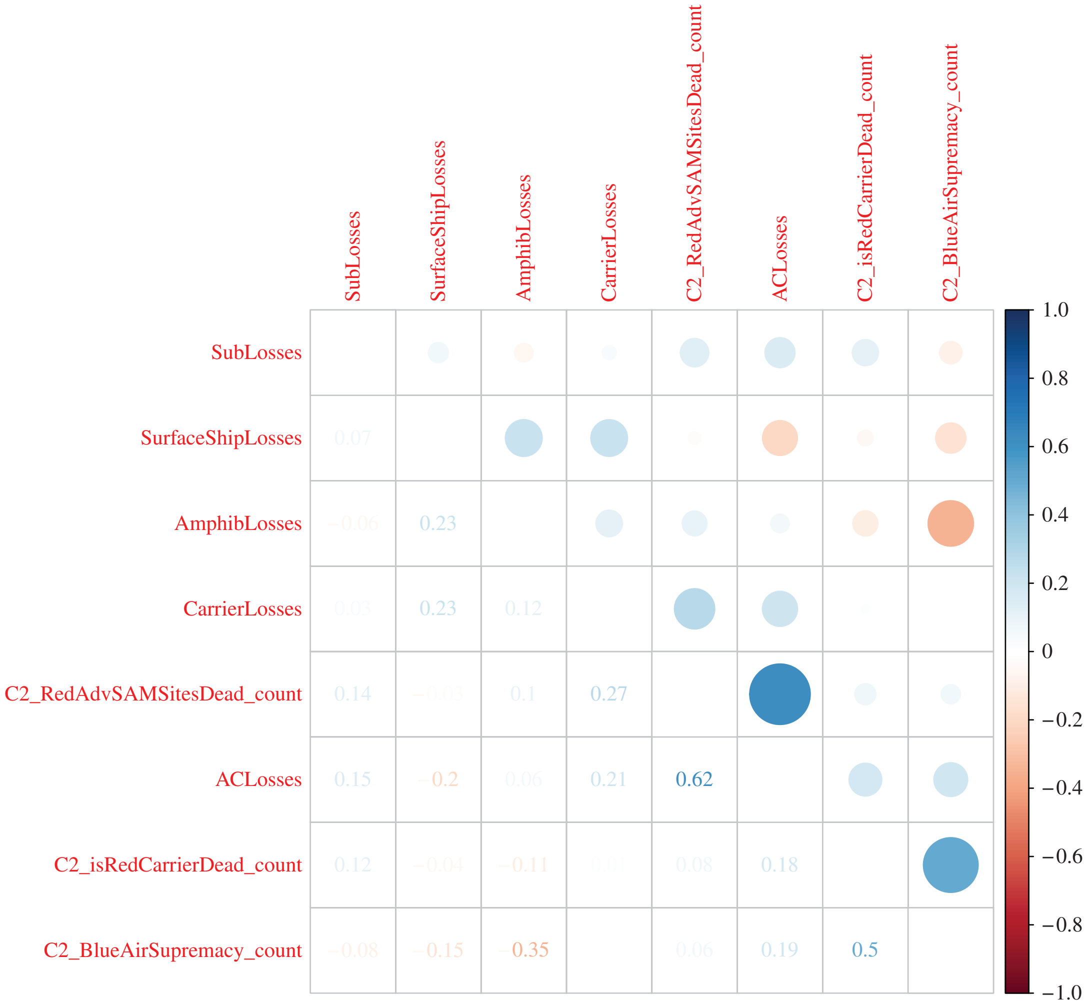

Criteria name links to metric's partition tree analysis
Metric's value at end of run
Replication number
Fail
Pass
In jeopardy

Variable	17	42	29	34	16	45	1	22	33	43	30	3	40	49	50	32	5	35	24	48	27	11	12	6	38	10	13	25	39	26	44	47	21	8	37	18	31	9	41	2	4	7	20	14	15	46	23	28	19	36
Blue_Carrier_Losses	1	0	0	1	0	1	1	1	1	1	0	0	1	0	2	1	1	1	1	1	3	2	2	2	2	0	2	2	1	1	2	1	3	3	3	3	3	2	2	3	2	3	2	2	2	3	2	3	3	1
Blue_SurfaceShip_Losses	8	6	4	4	9	6	8	6	7	9	5	7	6	6	9	9	9	10	7	9	9	4	7	8	7	7	9	6	11	8	8	8	6	8	7	7	9	5	8	10	6	7	10	10	5	8	7	11	9	10
Blue_Sub_Losses	3	5	4	4	4	4	5	4	5	5	3	4	3	4	4	4	3	4	4	4	4	4	3	4	5	5	5	5	5	5	5	5	6	5	4	5	3	4	4	4	4	4	5	4	4	3	6	4	4	5
Blue_Amphib_Losses	1	0	0	0	0	0	1	2	0	0	0	0	0	0	3	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	1
BlueAirSupremacy	0	4	4	9	5	2	0	1	0	5	13	6	6	0	0	0	6	0	0	6	12	10	13	6	9	8	6	7	0	9	15	17	4	4	0	0	6	0	0	3	5	5	5	10	6	5	0	0	6	0
BlueAirSuperiority	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
RedAirSupremacy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RedAirSuperiority	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
carthageESGHasEnteredMed	30	29	30	30	30	30	29	30	30	29	30	30	29	30	30	29	30	29	29	30	29	30	29	30	29	30	29	29	29	30	29	29	29	30	30	30	30	30	30	30	29	29	29	29	30	30	30	30	29	30
carthageCBGHasEnteredMed	30	30	30	30	31	30	30	31	31	30	30	30	30	30	30	30	31	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	31	31	31	30	30	31	30	30	30	30	30	30	30	31	30	30	30
carthageESGHasArrivedOffRome	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	10	6	6	6	6	6	7	11	9	6	6	8	6	6	6	6	6	6	12	6	6	6	9	6	6	6	6	6	6
expeditionaryOpsHaveBegun	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	9	5	5	5	5	5	5	9	7	5	5	7	5	5	5	5	5	5	11	5	5	5	8	5	5	5	5	5	5
carthageESGHasReturnedFromBeach	2	2	2	2	2	2	2	2	2	2	3	3	3	2	2	2	2	2	2	2	2	7	3	3	2	2	2	3	7	5	2	2	4	2	2	2	2	2	2	8	2	2	3	6	2	3	2	3	2	3
gibralterMinesweepingHasStarted	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
seabaseIsComplete	37	37	37	37	37	37	37	37	0	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	0	37	37	37	37
cruisersHaveArrivedAtSeabase	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37
seabaseNWScreenIsComplete	0	0	0	31	0	30	32	32	0	0	0	0	0	0	0	0	0	0	33	0	0	33	0	0	33	34	0	0	0	0	31	0	0	0	0	27	0	32	0	0	27	20	0	0	32	0	34	0	0	0
seabaseNEScreenIsComplete	37	37	37	37	37	37	37	37	0	37	37	0	0	37	37	37	37	0	37	37	37	37	37	37	37	37	37	37	0	37	37	37	37	37	37	37	0	37	37	37	37	37	37	37	37	37	37	0	0	37
isRedCarrierDead	0	6	0	0	0	0	4	5	0	0	0	6	0	0	0	4	0	0	0	6	12	18	13	6	9	8	6	7	17	17	15	11	0	0	0	0	6	10	0	0	0	0	0	0	6	6	0	0	3	0
seabaseSWScreenIsComplete	0	0	0	31	0	30	32	0	0	0	30	0	0	0	33	0	0	0	33	29	33	33	29	32	0	34	18	0	0	32	0	0	20	27	0	27	28	32	19	0	27	0	30	0	32	0	34	33	0	28
isESGInPort	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Color indicates which user-specified thresholds for success are met

Notes. The figure also shows the worst or best performance against these metrics. The replication numbers are not in numeric order because a clustering algorithm groups red cells together, making the dashboard easier to read by presenting less of a checkerboard display.

Figure 3. Correlation Plot of Key Metrics



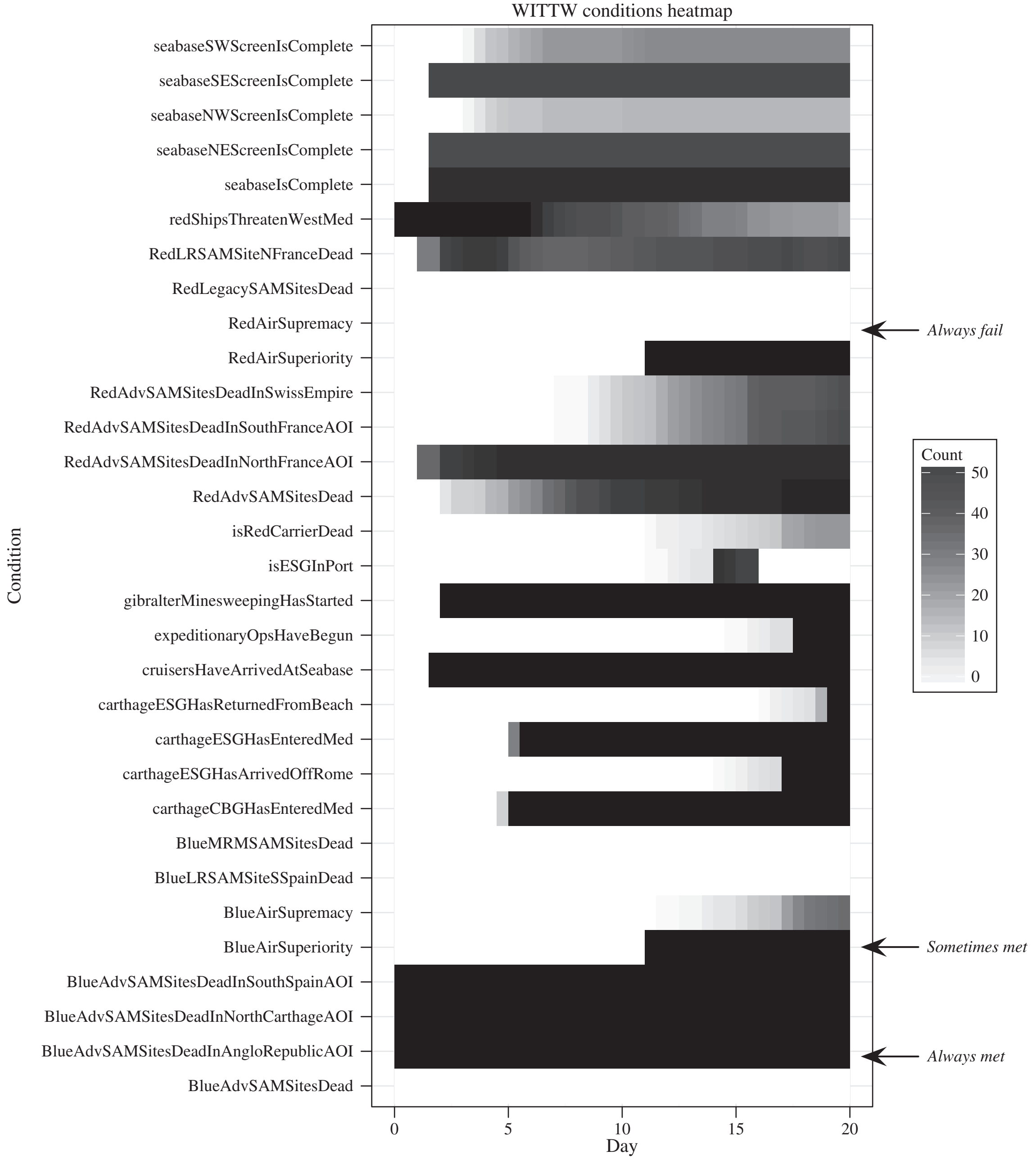
Notes. We take advantage of the fact that the correlation matrices are symmetric about the major diagonal to display the data numerically in the lower triangle and pictorially (color and size) in the upper triangle. Blue circles show positive correlations and red/orange circles show negative correlations.

several new heatmaps that allow analysts to identify if, when, and where user-specified conditions and events occurred. The conditions heatmap (Figure 4) displays the proportion of conditions that have occurred by campaign day for numerous user-defined conditions over the replications. For example, over all 50 replications, Red never achieves air supremacy during the first 20 days of the campaign. In contrast, Blue achieves air supremacy as early as Day 12, and with high probability by Day 18.

Event heatmaps reveal events in the simulation through space or time. STORMMiner creates several versions of the event heatmap. For example, Figure 5 shows the location of casualties broken down by platform type (e.g., by carrier, submarine, and surface ship) for both the Blue forces (top row) and Red forces (bottom row). The casualties heatmap can be tailored to multiple platform types and periods.

Figure 6 shows an example of a resource heatmap, which indicates status by type, time, and location. It can reveal useful patterns of resource levels that may, after further inspection, be attributed to consequences of earlier campaign progression (e.g., heavy use of ammunition during an engagement), or provide insight about later simulation states (e.g., reduced ability to prosecute targets during future periods). They visually indicate if,

Figure 4. Condition Heatmap Example



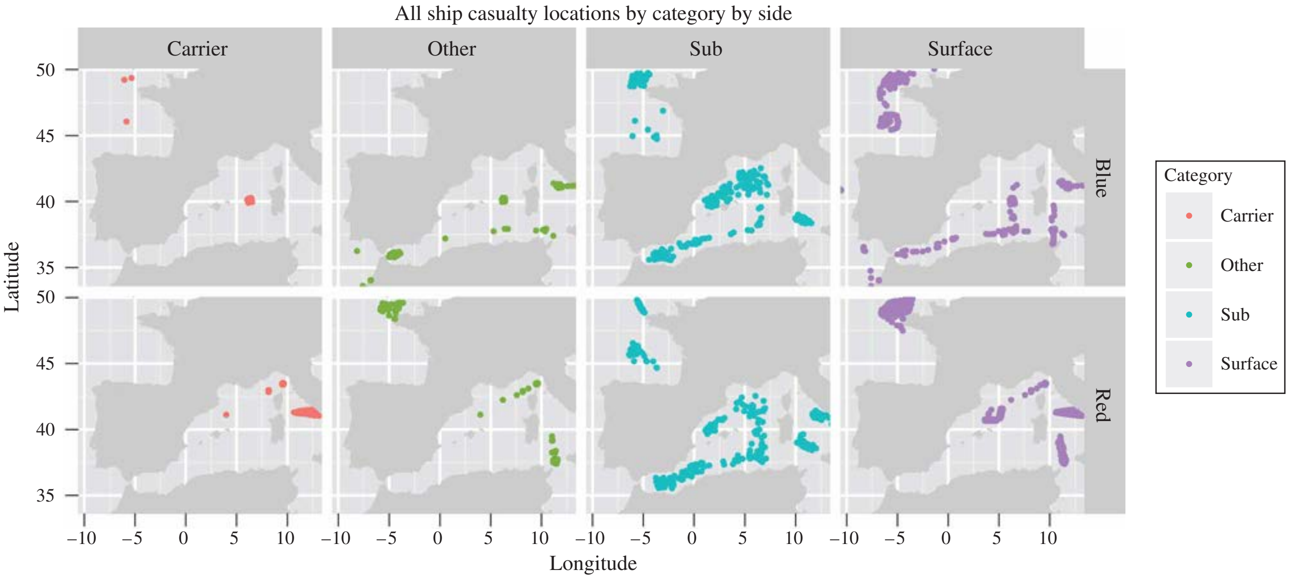
Notes. The vertical axis of the condition heatmap contains a row for every condition. The horizontal axis represents time, by day. For this run, we made 50 replications. White indicates that the condition was not true on that day for any of the 50 replications; black indicates that the condition was true on that day for all 50 replications; and gray indicates variability over the replications (sometimes true, sometimes not).

when, and where low resource levels are likely to occur. Other heatmaps track additional useful resources, such as the various petroleum, oil, and lubricant inventories; or other ammunition categories.

Cluster Analysis

Cluster analysis partitions the data into groupings (clusters) according to the 21 key WITTW metrics. This helps identify strong correlates among outcomes, allowing the analyst to focus on a few general but

Figure 5. Event Heatmap Example: A Colored Dot or Symbol Indicates Every Casualty That Takes Place in Any of the Replications



distinct ways the simulated campaign might unfold. The amount of separation between the clusters varies, depending on the nature of the data. We use `hclust`, R's hierarchical clustering function, implementing the clustering criteria of Ward (1963) via the `ward.D2` option, which "aims at finding compact, spherical clusters" (R documentation 2016). Figure 7 shows separation between two clusters, plotted using multidimensional scaling, in a form that is appealing to nontechnical audiences. The coordinate axes aggregate multiple, disparate measures, but the clusters are significantly different for nine of the 21 key metrics. For example, Cluster 1 has lower Blue losses across the board.

Another cluster analysis product (not shown) is a table that contains, for each key metric, the mean and standard deviation for each cluster, and the associated p-value from an analysis of variance. If the analyst determines that there are meaningful differences between the clusters, then the correlations plot, conditions heatmap, and other STORMMiner graphics can be produced by cluster.

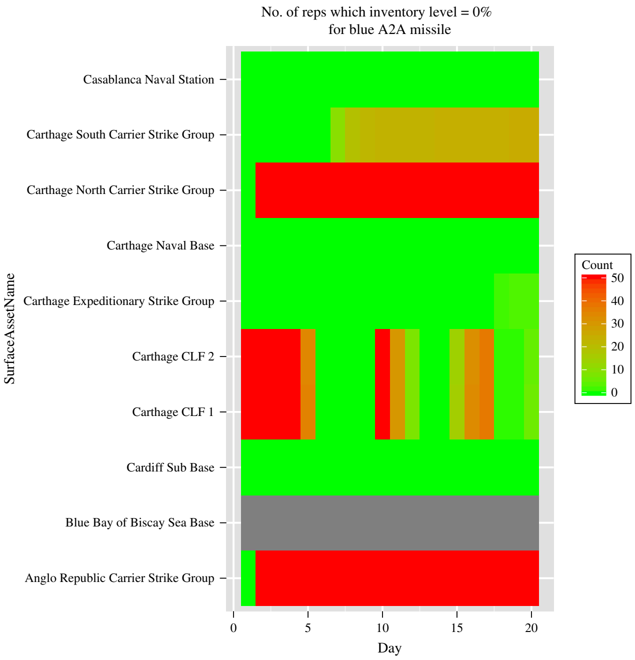
Unit and Plan Sequence Execution

Frequently, we are interested in the phased execution of the campaign plan. This is initially useful for troubleshooting during model development to ensure that all subcomponents of the model are functioning correctly. Ultimately, this provides insight into the potential evolution of the simulated campaign. In particular, this highlights the role of CONOPs and their relationship to the success and timing of various phases of the campaign.

Figure 8 shows two C2 plan sequence execution graphs for two different clusters. Note that the last two phases of the Carthage South carrier strike group (CSG) sometimes execute within 20 days for Cluster 1, where Blue does well; but never execute for Cluster 2, where Blue does poorly. Also note that only one of the execution plans is related to the overall campaign success—an insight that was heretofore unavailable. Recall that the C2 plans are not used to construct the clusters, and that they are but one of multiple components of the high-dimensional data that the analyst can explore for additional insight.

To investigate the reason for these phases sometimes not firing, we can look at Figure 9, which contains the unit C2 graph for the Carthage South CSG. The top-level purple node represents the major unit—in this case, the Carthage South CSG. The orange node contains the name of the plan sequence this graph represents—in this case the Carthage Med Plan. The blue nodes represent the phases of the plan. These phases are labeled A, B, C, D, and E in the C2 plan sequence execution graph (Figure 8). Inside the blue

Figure 6. Resource Heatmap

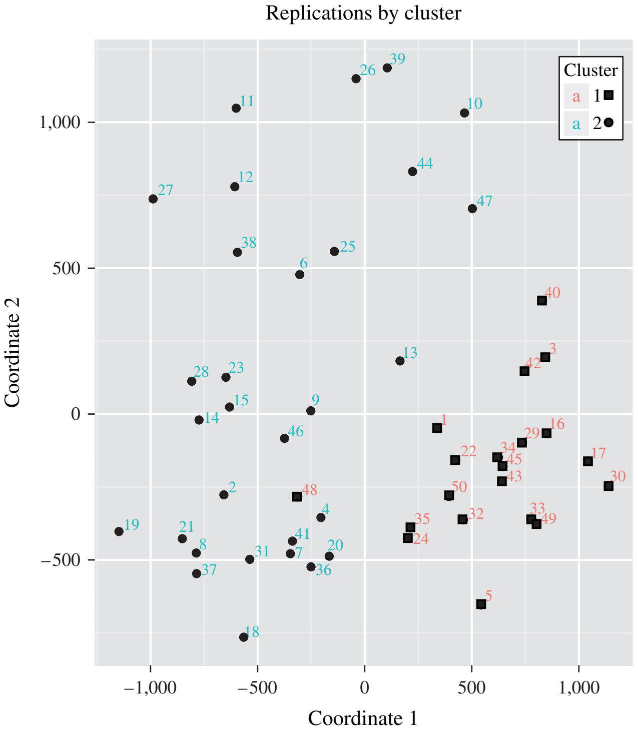


Notes. The horizontal axis represents time (i.e., days in the campaign). The maximum count in the legend indicates the number of replications (in this case, 50). The title of the graphic indicates which threshold value is being used. The user is provided with two heatmaps for each resource type—one to correspond with a " > 50%" threshold, and one to correspond with an "= 0%" threshold. The heatmap displays the status of resources for just the Blue air-to-air (A2A) missile, where the threshold corresponds to "= 0%." Here, the rows correspond to Blue assets that carry A2A missiles. The highest saturation of green corresponds to a resource level never (over all the replications) being equal to 0%. The highest saturation of red corresponds to a resource level always (over all the replications) being equal to 0%. The colors in between reflect variability.

nodes are the line numbers, from the naval orders file, which correspond to each plan of the plan sequence. For easy reference, we provide the mapping of unit plan phase text (e.g., "TRANSIT TO CENTRAL MED") to line numbers in a separate table. The pink nodes represent conditions that cause the unit to shift from one phase of execution to another. For example, there is an "UNTIL" condition at line number 876 that causes the Carthage South CSG to shift from Phase A to Phase B of its unit plan sequence. The yellow nodes exist to assign unique condition numbers to every user-defined condition (function) in the naval orders file, because a condition is often referenced by more than one unit.

The grayscale nodes that appear next to the pink condition nodes are execution nodes, and contain information about the firing of the different phases of the unit plan. For example, in Figure 9, the execution node for Phase A (line number 873 in the orders file) contains 0.00 followed by 50. This means that this plan phase executed in all 50 replications, with a median firing time of zero (scenario start). The execution node for Phase E (line number 909), however, fired only five out of 50 times, with a median execution time of 16.88. In this example, Phase D fired only once out of the 50 replications; it was skipped during four out of five runs where Phase E fired. The color gray corresponds with the proportion of times (over the replications) that the phase executed, with darker colors corresponding to higher proportions.

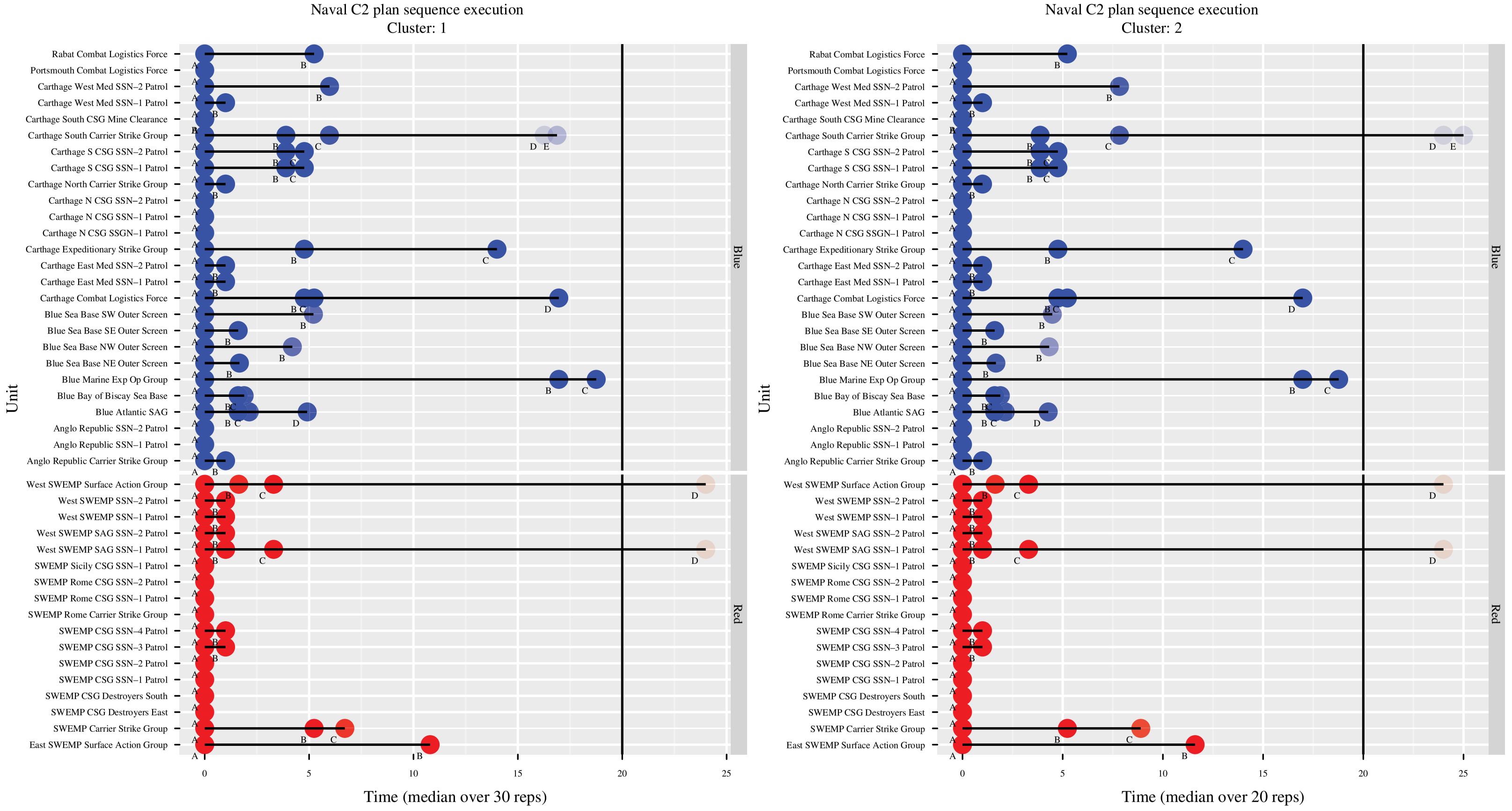
Figure 7. Multidimensional Scaling Depiction of the Separation Between Two Clusters, Where the Clusters Are Determined by WITTW Key Metrics



Metamodel-Based Approaches

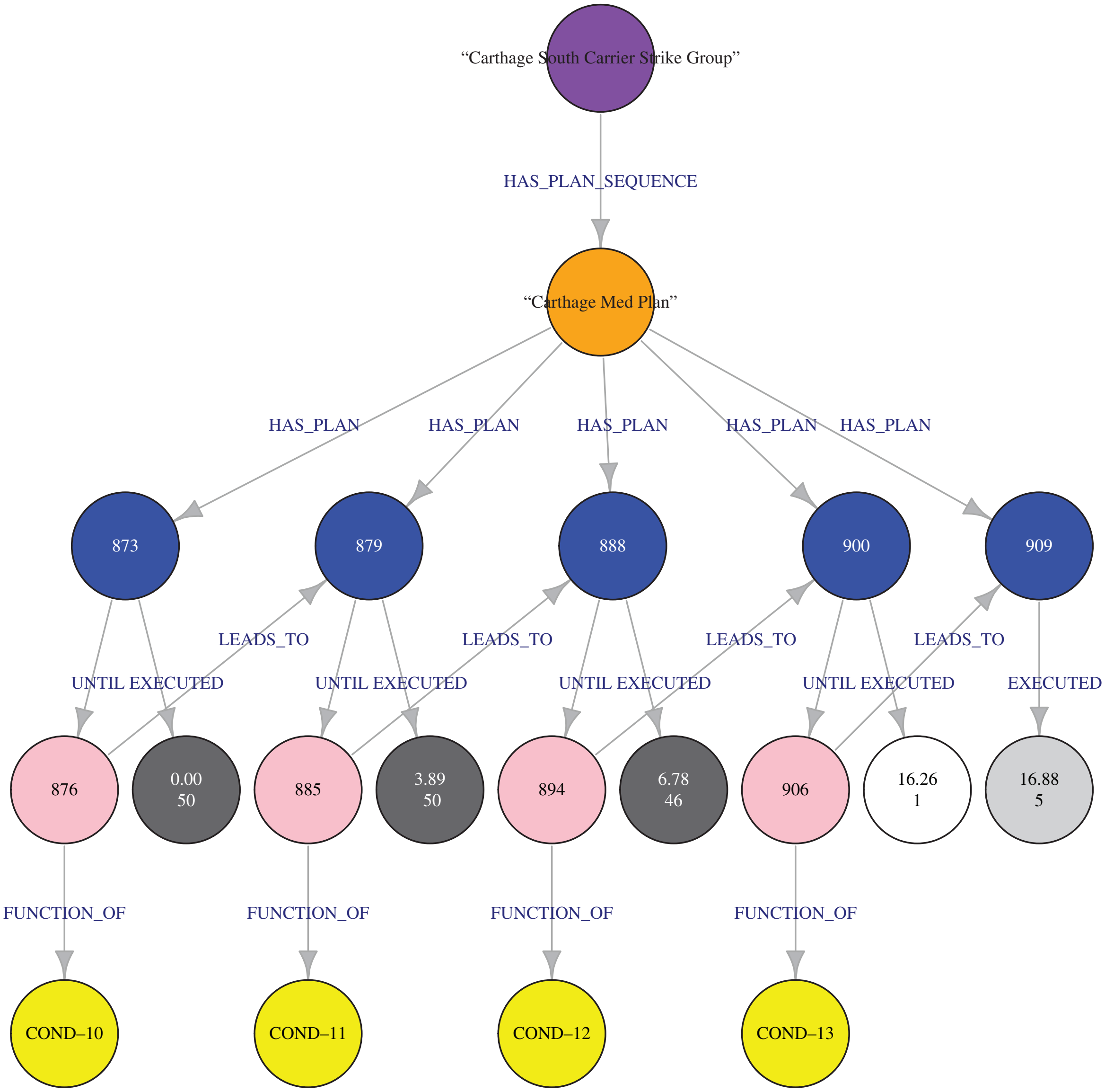
A common approach to help analysts glean insights from simulation output involves fitting statistical metamodels. Metamodels are mathematical models that encapsulate the observed behavior of the simulation by mapping inputs to outputs. While simulation runs may require a massive setup effort and have long run times,

Figure 8. A Side-by-Side Comparison of Two C2 Plan Sequence Execution Graphs Shows the Difference (by Cluster)



Note. In this example, the Carthage South carrier strike group is significantly delayed in Cluster 2 compared with Cluster 1.

Figure 9. The C2 Graph for the Carthage South Carrier Strike Group (CSG): From This Graph, We Obtain Information About the Execution (Firing) of Major Phases of the Unit Plan (Blue Nodes, Level 3) and the Conditions That Trip Those Phases of the Plan (Pink Nodes, Level 4)



Notes. The conditions that trip those phases of the plan (pink nodes). The grayscale nodes give execution data: the median time that phase of the plan fired, and the total number of replications for which that phase of the plan fired at all. Transparency indicates fewer instances of that phase of the plan being executed.

metamodels facilitate quick-turn, exploratory analysis (Kleijnen 2015, Rosen et al. 2015).

To date, we have focused on creating and describing metamodel-based tools that help analysts gain similar types of insights by mapping intermediate observations (simulation states) to campaign outcomes, where the variation in intermediate states is due to stochastic model components. For example, we do not explicitly explore how the initial number of Blue carriers affects Blue campaign outcomes. However, we can observe the correlation between early Blue carrier losses and subsequent campaign outcomes across multiple replications of this scenario. This approach allows the analyst to investigate a plethora of intermediate outcomes from a limited number of simulation runs using this type of metamodel-based approach. This may provide useful insights, reveal unanticipated relationships, or raise questions that require more detailed research and assessment.

One valuable summary tool we built creates a random forest of partition trees, as Kuhn and Johnson (2013) describe, using the `rpart` and `randomForest` packages from R. The analyst is presented with a variable importance graph that provides a relative ranking of influential variables (intermediate states), a set of partial dependence plots, the best partition tree, and the contour plot displaying the top-two influential variables.

It is possible—likely even—that partition trees produce collinear candidate variables when the data arise from stochastic variation over replications of a single scenario (excursion, or design point in design-of-experiments methodology). This is a feature of the type of problems analyzed in campaign analysis and cannot be removed through postprocessing. Using designed experiments for structured variation of input data would permit analysts to determine cause-and-effect relationships within the model. Ultimately, all these automated postprocessing tools are used to enable analyst insight.

As previously mentioned, the examples in this section arise from a notional training scenario. To customize to a specific campaign, an analyst must first generate specific artifacts that STORMMiner requires to postprocess. We provide guidance and scripts to facilitate this effort. We performed this customization process for one classified scenario, and the entire customization process took only about 30 minutes.

Summary

The U.S. Navy's OR team is charged with providing top-level decision makers with the information they need to shape the Navy for the foreseeable future. These are high-impact decisions that affect our national security. To support this effort, the Navy has a large in-house cadre of highly educated OR professionals, both military and civilian, and leverages the expertise of both public and private sector analysts. Accordingly, the Navy directly invests tens of millions of dollars annually in simulation modeling and analysis. It is imperative that the senior leaders, analysts, and other stakeholders have the tools and processes they need—embedded in a effective management structure—to obtain the best possible information within the required timeline.

Campaign simulation models produce large volumes of data from which insights must be extracted; thus, new and innovative big data tools and analysis approaches are required. These tools need to be sufficiently rich to be useful to the analysts who work in the day-to-day development of the studies, but succinct and accessible to be used by leaders (generally, but not always, without analytical backgrounds) who will ultimately make investment decisions.

Since we implemented these technical and management controls, we have enjoyed significant improvement in the execution, performance, and quality of our simulation analyses. We believe that these lessons—particularly treating large-scale simulation as a software-engineering task—are scalable and immediately applicable to other organizations both in and out of the defense arena. Next, we describe some specific achievements.

First, this approach has dramatically reduced the time and costs of studies. By defining the analytic questions and modeling requirements up front and then mapping them to a spiral development process, our studies are now approximately 33 percent faster and 16 percent less expensive. These methods have also increased our analytic maturity across the organization.

Second, these new and advanced data analysis tools have enabled analysts to identify trends, quantify relationships, and create insights that heretofore were very difficult and time consuming to detect, explain, and defend. This new knowledge greatly improves our risk assessments and development of recommendations. Moreover, these tools improve the testing and vetting of studies.

Third, the insights and recommendations generated are more credible because they involve a transparent development process that undergoes constant review and testing by both military and civilian analysts, who apply their judgment, experience, and operational expertise on how to best communicate key findings to leadership. While not all of the products generated by STORMMiner are meant to be put in front of leadership, some of them can be. This enables decision makers to see smooth, stylized briefing materials that can be supplemented with elements directly from the simulation suite.

Finally, this approach has improved cooperation and coordination among all stakeholders. The new tools allow us to identify complex trends and hidden relationships in a timely manner, while the definitive schedule ensures that we meet our partners' expectations and needs. The deliberate co-development of pre-analysis tasks, which focus on planning and management, alongside postdevelopment tasks, which focus on output processing, has rapidly increased the maturity of our organization in a way that focus on one or the other alone could not.

The greatest contribution from our effort, both from a management and big data perspective, has been the delivery of better products on predictable timelines to enable better decisions. Although our use case has been focused on a large-scale military campaign scenario, we believe this general approach has application to many other enterprise-level, risk-based decision-making endeavors.

Acknowledgments

The authors thank Ms. Virginia "Robbin" Beall of the Senior Executive Service for her support and guidance in this and other projects. The authors also thank the U.S. Navy's Naval Research Program at NPS for funding the SEED Center portion of this research. Department of Defense Distribution Statement: Approved for public release; distribution is unlimited. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Navy, Department of Defense, or Government.

References

Barber AH (2014) Joint warfare analysis—The key to shaping DoD's future. *Phalanx* 47(1):50–52.

Beall VR (2015) Defense innovation through wargaming. *Phalanx* 48(3):58–60.

Bickel WG Jr (2014) Improving the analysis capabilities of the Synthetic Theater Operations Research Model (STORM). Master's thesis, Naval Postgraduate School, Monterey, CA. Accessed February 10, 2017, http://calhoun.nps.edu/bitstream/handle/10945/43878/14Sep_Bickel_William.pdf?sequence=1.

Boehm B (2000) Spiral development: Experience, principles, and refinements. Technical Report CMU/SEI-2000-SR-008, Carnegie Mellon University, Pittsburgh, PA.

Bracken J, Kress M, Rosenthal RE (1995) *Warfare Modeling* (John Wiley & Sons, New York).

Griffith SB (1963) *The Art of War* (Oxford University Press, New York).

Group W (2012a) *STORM: Analyst's Manual Version* 2.3 (Group W, Fairfax, VA).

Group W (2012b) *STORM: Programmer's Manual Version* 2.3 (Group W, Fairfax, VA).

Group W (2012c) *STORM: User's Manual Version* 2.3 (Group W, Fairfax, VA).

Hodges JS (1991) Six (or so) things you can do with a bad model. *Oper. Res.* 39(3):355–365.

Hughes WP Jr. (1999) *Joint Campaign Analysis Book I—Student Text* (Naval Postgraduate School, Monterey, CA). Accessed February 10, 2017, https://calhoun.nps.edu/bitstream/handle/10945/43201/JCABook1.pdf.

Kleijnen JPC (2015) *Design and Analysis of Simulation Experiments*, 2nd ed. (Springer, New York).

Kline JE, Hughes WP Jr., Otte D (2011) Campaign analysis: An introductory review. Cochran J, ed. *Wiley Encyclopedia of Operations Research and Management Science* (John Wiley & Sons, New York).

Kress M, Talmor I (1999) A new look at the 3:1 rule of combat through Markov stochastic Lanchester models. *J. Oper. Res. Soc.* 50(7): 733–744.

Kuhn M, Johnson K (2013) *Applied Predictive Modeling* (Springer, New York).

Lanchester FW (1916) *Aircraft in Warfare: The Dawn of the Fourth Arm* (Constable and Company Limited, London).

Loerch AG, Rainey LB (2007) *Methods for Conducting Military Operational Analysis* (Military Operations Research Society, Arlington, VA).

Lucas TW, Kelton WD, Sánchez PJ, Sanchez SM, Anderson BL (2015) Changing the paradigm: Simulation, now a method of first resort. *Naval Res. Logist.* 62(4):293–305.

McDonald ML, Upton SC, Seymour CN, Lucas TW, Sanchez SM, Sanchez PJ, Schramm HC, Smith JR (2014) Enhancing the analytic utility of the Synthetic Theater Operations Research Model (STORM). Tolk A, Diallo SY, Ryzhov O, Yilmaz L, Buckley S, Miller JA, eds. *Proc. 2014 Winter Simulation Conf.* (IEEE, Piscataway, NJ), 4136–4137.

Morse P, Kimball G (1951) *Methods of Operations Research* (Technology Press of MIT, Cambridge, MA & John Wiley, New York).

R documentation (2016) *R Documentation: Hierarchical clustering*. Accessed February 10, 2017, https://stat.ethz.ch/R-manual/R-devel/library/stats/html/hclust.html.

Rosen SL, Saunders CP, Gurahay S (2015) A structured approach for rapidly mapping multilevel system measures via simulation metamodeling. *Systems Engrg.* 18(1):87–101.

Saeger K, Hinch J (2001) Understanding instability in a complex deterministic combat simulation. *Military Oper. Res.* 6(4):43–55.

Sanchez SM (2015) Simulation experiments: Better data, not just big data. Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD, eds. *Proc. 2015 Winter Simulation Conf.* (IEEE, Piscataway, NJ), 800–811.

Sanchez SM, Lucas TW, Sanchez PJ, Nannini CJ, Wan H (2012) Designs for large-scale simulation experiments, with applications to defense and homeland security. Hinkelmann K, ed. *Design and Analysis of Experiments: Special Designs and Applications*, 1st ed., v3 (John Wiley & Sons, New York), 413–441.

SEED Center for Data Farming (2015) *STORMMiner User's Manual, Version* 1.0 (Naval Postgraduate School, Monterey, CA).

Seymour CN (2014) Capturing the full potential of the Synthetic heater Operations Research Model (STORM). Master's thesis, Naval Postgraduate School, Monterey, CA. Accessed February 10, 2017, http://calhoun.nps.edu/bitstream/handle/10945/44000/14Sep_Seymour_Christian.pdf?sequence=1.

Singham DI (2014) Selecting stopping rules for confidence interval procedures. *ACM Trans. Model. Comput. Simulation* 24(3):Article no. 18.

United States Office of Management and Budget (2016) *Fiscal Year* 2016 *Budget of the U.S. Government*. Accessed February 10, 2017, https://www.gpo.gov/fdsys/pkg/BUDGET-2016-BUD/pdf/BUDGET-2016-BUD.pdf.

Wagner DH, Mylander W, Sanders T (1999) *Naval Operations Analysis* (Naval Institute Press, Annapolis, MD).

Ward JH (1963) Hierarchical grouping to optimize an objective function. *J. Amer. Statist. Assoc.* 58(30):236–244.

Washburn AR, Kress M (2009) *Combat Modeling* (Springer, New York).

Witten IH, Frank E, Hall MA (2011) *Data Mining: Practical Machine Learning Tools and Techniques* (Elsevier Science, Burlington, MA).

Verification Letter

Robbin Beall, Senior Executive Service, Branch Head, Campaign Analysis and Modeling, Navy Staff, Washington, DC 20350, writes:

“As required by your submission policy, I verify that the technical approaches and advances as explained in the submitted article ‘US Navy Uses Big Data Approaches to Improve Decisions’ are correct.

“I have been the senior representative from our analysis organization present when at various times the analysis this project improves was presented to Chief of Naval Operations, Director, Office of Net Assessment, and Undersecretary of the Air Force, as well as numerous other audiences of our most senior civilian and military leaders. The analysis work that this team performs guides decisions in all aspects of the US Navy’s strategy and budget.

“I highly encourage *Interfaces* to publish this article.”

CAPT Brian L. Morgan served as the Navy’s senior military analyst and Campaign Analysis and Modeling Deputy Branch Head while assigned to the Assessment Division (OPNAV N81). He is Operations Research Program Officer and a senior lecturer at the Naval Postgraduate School. He holds an MS in operations research from the Naval Postgraduate School and a BS from the University of Virginia, and serves on the MORS Board of Directors.

Harrison C. Schramm is a Principal Operations Research Analyst at CANA Advisors. While on Active Duty with the U.S. Navy, he served as the Campaign Analysis Section Head. He holds an MS in operations research from the Naval Postgraduate School and a bachelor’s degree from the U.S. Naval Academy, and is a past recipient of the MORS Barchi prize. He enjoys professional accreditation from INFORMS (CAP) and the ASA (PStat).

Jerry R. Smith, Jr. is a civilian operations research analyst for the Navy. He holds degrees in physics, engineering, and systems analysis. From 2008 to 2016, he led the World Class Models (WCM) initiative that improved the Navy’s decision tools, business analysis methods, and combat simulations in manpower, maintenance, operations, warfighting assessments, and new aircraft/ship concept designs. He is on the Board of Directors for the Military Operations Research Society and is a member of the Acquisition Professional Community.

Thomas W. Lucas is a professor in the Operations Research Department at the Naval Postgraduate School (NPS), where he joined the faculty in 1998. In 2006 he co-founded the SEED Center for Data Farming, and continues to be its co-director. Previously, he worked as a statistician and project leader for six years at RAND and as a systems engineer for 11 years at Hughes Aircraft Company. His primary research interests include military modeling and analysis, high-dimensional design of computational experiments, and robust Bayesian statistics.

Mary L. McDonald is a research associate for the SEED Center for Data Farming at the Naval Postgraduate School, where she supports projects that advance the development and use of simulation experiments and efficient design. She has a BA in mathematics from Northwestern University and an MS in applied mathematics from the Naval Postgraduate School. She teaches probability and statistics and provides guest lectures for modules on combat modeling. Her research interests include evolutionary computation, high-dimensional data analysis, and agent-based simulation.

Paul J. Sánchez is a faculty member in the Operations Research Department at the Naval Postgraduate School. He has an BS in economics from MIT, and an MS and PhD in operations research from Cornell. His research interests include the design and analysis of large-scale simulation experiments, robust design, and object-oriented computational modeling, with application to military operations and healthcare. He is an active member of the NPS SEED Center for Data Farming.

Susan M. Sanchez is a professor in the Operations Research Department at the Naval Postgraduate School, with a joint appointment in the Graduate School of Business and Public Policy. She has a BSE in industrial and operations engineering from the University of Michigan, and an MS and PhD in operations research from Cornell. Her research interests include the design and analysis of large-scale simulation experiments, robust design, and applied statistics, with application to military operations, manufacturing, and healthcare. She co-founded and serves as co-director of NPS’ SEED Center for Data Farming.

Stephen C. Upton is a research associate for the SEED Center for Data Farming at the Naval Postgraduate School, where he supports projects that advance the development and use of simulation experiments and efficient designs. He has a BS in physics from the University of Idaho, MS in operations research and physics from the Naval Postgraduate School, and an applied scientist in systems engineering from the George Washington University. He worked as a research scientist for Referentia Systems, the MITRE Corporation, and the Los Alamos National Laboratory, and served 24 years on Active Duty with the U.S. Marine Corps.