RAND

REPRINTS

RP-925

Exploratory Analysis and a Case History of Multiresolution, Multiperspective Modeling

Paul K. Davis, James H. Bigelow, and Jimmie McEver

Reprinted from *Proceedings of the 2000 Winter Simulation Conference, Jeffrey A. Joines, Russell R. Barton, K. Kang, and* Paul A. Fishwick (editors), December, 2000 and *Proceedings of the SPIE, Vol. 4026, 2000*

National Defense Research Institute

Preface

The papers reprinted here originated in an effort to illustrate concretely how modeling and analysis could and should be used to help inform efforts by the Department of Defense to understand and evaluate advanced concepts and define critical issues worthy of experimentation. A core element of our approach was the notion that analysis should use *families* of models and war games. It should not depend too heavily on work at any one level of resolution or with any single perspective of how to "look at" the issues. For our prototype problem we chose the challenge of interdicting and halting an invading army through the use of long-range precision fires delivered from aircraft and either sea-based or ground-based missiles.

The first paper in this reprint volume describes the important concept of *exploratory analysis*, which seeks to gain a broad understanding of a problem domain before going into details for particular cases. For this one needs suitable low-resolution models that can be readily understood. The paper then explains how systematic exploratory analysis is enabled by multiresolution, multiperspective modeling, which is becoming increasingly feasible for reasons discussed in the paper.

The second paper of the series describes how we informed and calibrated our low-resolution exploratory-analysis models by drawing upon higher-resolution information, including results of high-resolution simulation experiments that tracked tanks and other entities over time. These experiments generated "data" that we analyzed in depth much as though it were empirical in nature. To do the analysis we constructed a bridging model, which was was also relatively simple, but with multiple levels of resolution and perspective. This model was motivated by the physics of the problem as we understood it. After extensive data analysis, we ended up adjusting some of the model's basic concepts, calibrating variables where we believed doing so was meaningful, and noting other effects that could not properly be calibrated and should be treated as stochastic or otherwise uncertain. We are now able not only to understand the original high-resolution results, but to make reasonable estimates of how the effectiveness of precision fires would vary with factors such as the size and character of the invading force, the movement rate of that force, the character of the terrain in which interdiction is attempted, the command and control effectiveness of interdicting forces and weapons, key characteristics of the weapons themselves, and, of course, the size of the interdiction force. We are able to show that the effectiveness of some state-of-the-art fires should be expected to vary by more than two orders of magnitude, depending on these factors.

The third paper provides an overview of how the different levels and styles of modeling related to each other, and what lessons we learned technologically and theoretically about activities of this sort. Some of the lessons have implications for toolbuilding--i.e., for enabling technology that will, in the future, facilitate: exploratory analysis; multiresolution, multiperspective modeling; and mutual calibration with "families" of models and games.

Our work was conducted as part of a special cross-cutting project by RAND's National Defense Research Institute (NDRI), a federally funded research and development center sponsored by the Office of the Secretary of Defense, the Joint Staff, unified commands, and defense agencies. The principal sponsor was Mr. James Johnson, Deputy Director (Theater Assessments and Planning) of the Office of Program Analysis and Evaluation in the Office of the Secretary of Defense.

Exploratory Analysis Enabled by Multiresolution, Multiperspective Modeling

Paul K. Davis

Reprinted from *Proceedings of the 2000 Winter Simulation Conference, Jeffrey A. Joines, Russell R. Barton, K. Ka*ng, and Paul A. Fishwick (editors). A preliminary version appeared in Proc. SPIE Vol. 4026, p. 2-15, Enabling Technology for Simulatin Science IV, Alex F. Sisti; Ed., June, 2000

National Defense Research Institute

EXPLORATORY ANALYSIS ENABLED BY MULTIRESOLULTION, MULTIPERSPECTIVE MODELING

Paul K. Davis

RAND and the RAND Graduate School
1700 Main St., Santa Monica, CA 90407-2138, U.S.A.

ABSTRACT

The objective of exploratory analysis is to gain a broad understanding of a problem domain before going into details for particular cases. Its focus is understanding comprehensively the consequences of uncertainty, which requires a good deal more than normal sensitivity analysis. Such analysis is facilitated by multiresolution, multiperspective modeling (MRMPM) structures that are becoming increasingly practical. A knowledge of related design principles can help build interfaces to more normal legacy models, which can also be used for exploration.

1 BACKGROUND

Strategy problems are typically characterized by enormous uncertainties that should be central in assessment of alternative courses of action—although individuals and organizations often suppress those uncertainties and give a bizarre level of credence to wishful-thinking planning factors and other simplifications (Davis 1994 Ch. 4, Davis, Gompert, and Kugler 1996). In the past, an excuse for downplaying uncertainty analysis—except for marginal sensitivity analysis around some "best-estimate" baseline of dubious validity—was the sheer difficulty of doing better. The time required for setup, run, and analysis made extensive uncertainty work infeasible. Today, technology permits extensive uncertainty analysis with personal computers.

A key to treating uncertainty well is *exploratory analysis* (Davis and Hillestad 2001). The objectives of exploratory analysis include understanding the implications of uncertainty for the problem at hand and informing the choice of strategy and subsequent modifications. In particular, *exploratory analysis can help identify strategies that are flexible, adaptive, and robust*. In successive sections, this paper describes exploratory analysis; puts it in context; discusses enabling technology and theory; points to companion papers applying the ideas; and concludes with some technology challenges for modeling and simulation. The paper draws heavily on a forthcoming book (Davis and Hillestad 2001) and builds on a much rougher preliminary presentation of the same material (Davis 2000).

2 EXPLORATORY ANALYSIS

2.1 What Exploratory Analysis Is and Is Not

Exploratory analysis examines the consequences of uncertainty. It can be thought of as sensitivity analysis done right, but is so different from usual sensitivity analysis as to deserve a separate name. It is closely related to scenario space analysis (Davis 1994 Ch. 4) and "exploratory modeling" (Bankes 1993, Lempert et al. 1996). It is particularly useful for gaining a broad understanding of a problem domain before dipping into details. That, in turn, can greatly assist in the development and

---

Reprinted by permission from *Proceedings of the 2000 Winter Simulation Conference*, Jeffrey A. Joines, Russell R. Barton, K. Kang, and Paul A. Fishwick, editors. (A preliminary version appeared in *Proceedings of SPIE*, Vol. 4026, pp. 2-15, "Enabling Technology for Simulation Science IV," Alex F. Sisti, editor, June 2000.)

choice of strategies. It can also enhance "capabilities-based planning" by clarifying *when*—i.e., in what circumstances and with what assumptions about all the other factors—a given capability such as an improved weapon system or enhanced command and control will likely be sufficient or effective (Davis, Gompert, and Kugler 1996). This contrasts with establishing a base-case scenario, and an organizationally blessed model and data base, and then asking "How does the outcome change if I have more of this capability?"

2.2 Types of Uncertainty

Uncertainty comes in many forms and it is useful (National Research Council 1997) to distinguish between input uncertainties (i.e., parametric uncertainties) and structural uncertainty. Input uncertainty relates to imprecise knowledge of the model's input values. Structural uncertainty relates to questions about the form of the model itself: Does it reflect all the variables on which the real-world phenomenon purportedly described by the model depends? Is the analytical form correct? Some uncertainties may be inherent because they represent stochastic processes. Some may relate to fuzziness or imprecision, while others reflect discord among experts. Some relate to knowledge about the values of well-defined parameters, whereas others refer to future values that as yet have no true values.

It is convenient to express the uncertainties parametrically. If unsure about the model's form, we can describe this also to some extent with parameters. For example, parameters may control the relative size of quadratic and exponential terms in an otherwise linear model. Or a discrete parameter may be a switch choosing among distinct analytical forms. Some parameters may apply to the deterministic aspect of a model, others to a stochastic aspect. For example, a model might describe the rate at which Red and Blue suffer attrition in combat according to a simplistic Lanchester square law:

$$\frac{d\tilde{R}}{dt} = -\tilde{K}_b\tilde{B}(t) \quad \frac{d\tilde{B}}{dt} = -\tilde{K}_r\tilde{R}(t)$$

where the attrition coefficients for Red and Blue have both deterministic and stochastic parts, each of which are subject to uncertainty, as in (illustrating for Blue only)

$$\tilde{K}_b(t) = K_{bo}[1 + c_b\tilde{N}_b(t;\mu,\sigma_b)].$$

Here the N term is a normal random variable with mean μ and standard deviation σ. It represents stochastic processes occurring within a particular simulated war, e.g., from one time period to the next. The means and standard deviations are ordinary deterministic parameters, as are the coefficients K_{bo}, K_{ro}, c_r, and c_b. These have constant values within a particular war, but at what value they are constant is uncertain.

So far the equations have represented input uncertainty. However, suppose there is controversy over using the linear, square, or some hybrid version of a Lanchester equation. We could represent this dispute as input, or parametric, uncertainty by modifying the equation to read

$$\frac{d\tilde{R}}{dt} = -\tilde{K}_b\tilde{B}^e(t)\tilde{R}^f(t) \quad \frac{d\tilde{B}}{dt} = -\tilde{K}_r\tilde{B}^g(t)\tilde{R}^h(t).$$

Now, by treating the exponents as uncertain parameters, we could explore both input and structural uncertainties in the model—at least to some extent. The fly in the ointment is that nature's combat equations are much more complex (if they exist), and we don't even know their form. Suppose, merely as an example, that combatant effectiveness decays exponentially as combatants grow weary. We could not explore the consequences of different decay times if we did not even recognize the phenomenon in the equation's form. In fact, we *often* do not know the true system model. Nonetheless, much can be accomplished by allowing for diverse effects parametrically.

1.3 Types of Exploratory Analysis

Exploratory analysis can be conducted in several ways (Davis and Hillestad 2001). Although most of the methods have been used in the past (see especially Morgan and Henrion 1992), they are still not appreciated and are often poorly understood.

Input exploration (or *parametric exploration*) involves conducting model runs across the space of cases defined by discrete values of the parameters within their plausible domains. It considers not just excursions taken one-at-a-time as in normal sensitivity analysis relative to some presumed base-case set of values, but rather all the cases corresponding to value combinations defined by an experimental design (or a smaller sample). The results of such runs, which may number from dozens to hundreds of thousands or more, can be explored interactively with modern displays. Within perhaps a half-hour, a good analyst doing such exploration can often gain numerous important insights that were previously buried. He can understand not just which variables “matter,” but *when*. For example, he may find that the outcome of the analysis may be rather insensitive to a given parameter for the so-called base case of assumptions, but quite sensitive for other plausible assumptions. That is, he may identify in what cases the parameter is important. To do capabilities-based planning for complex systems, this can be distinctly nontrivial.

A complement to parametric exploration is "*probabilistic exploration*" in which uncertainty about the input parameters is represented by distribution functions representing the totality of one’s so-called objective and subjective knowledge. I sometimes use quotes around "probability" because the distributions are seldom true frequencies or rigorous Bayesian probabilities, but rather rough estimates or analytical conveniences.

Using analytical or Monte Carlo methods, the resulting distribution of outcomes can be calculated. This can quickly give a sense for whether uncertainty is particularly important. In contrast to displays of parametric exploration, the output of probabilistic exploration gives little visual weight to improbable cases in which various inputs all have unlikely values simultaneously. Probabilistic exploration can be very useful for a condensed net assessment. Note that this use of probability methods is different from using them to describe the consequences of a stochastic process within a given simulation run. Indeed, one should be cautious about using probabilistic exploration because one can readily confuse variation across an ensemble of possible cases (e.g., different runs of a war simulation) with variation within a single case (e.g., fluctuation from day to day within a single simulated war). Also, an unknown constant parameter for a given simulated war is no longer unknown once the simulation begins and simulation agents representing commanders should perhaps observe and act upon the correct values within a few simulated time periods. Despite these subtleties, probabilistic exploration can be quite helpful.

The preferred approach treats some uncertainties parametrically and others with uncertainty distributions. That is, it is *hybrid exploration*. It may be appropriate to parameterize a few key variables that are under one’s own control (purchases, allocation of resources, and so on), while treating the uncertainty of other variables through uncertainty distributions. One may also want also to parameterize a few variables characterizing the future context in which strategy must operate (e.g., short warning time). There is no general procedure here; instead, the procedure should be tailored to the problem at hand. In any case, the result can be a comprehensible summary of how known classes of uncertainty affect the problem at hand.

Let me give a few examples of what exploratory analysis can look like. Figure 1 mimics a computer screen during a parametric exploration of what is required militarily to defend Kuwait against a future Iraqi invasion by interdicting the attacker’s movement with aircraft and missiles (Davis and Carrillo 1997). Each square denotes the outcome of a particular model case (i.e., a specific choice of all the input values). The model being used depends on 10 variables–those on the x, y, and z axes, and seven listed to the side (the z–axis variable is also listed there, redundantly). The outcome of a given simulation is represented by the color (or, in this paper, by the pattern) of a given square. Thus, a white square represents a good case in which the attacker penetrates only a few tens of kilometers before being halted. A black square represents a bad case in which the attacker penetrates deep into the region that contains critical oil facilities. The other patterns represent in-between cases. The number in each square gives the penetration distance in km.

To display results in this way for a sizable scenario space RAND has often used a program called Data View, developed at RAND in the mid 1990s by Stephen Bankes and James Gillogly. After running the thousands or hundreds of thousands of cases corresponding to an experimental design for parametric exploration, we explore the outcome space at the computer. We can choose interactively which of the parameters to vary along the x, y, and z axes of the display. The other parameters then have the values shown along the right. However, we can click on their values and change them interactively by selecting from the menu of values for cases that have been run.

As mentioned above, in about a half an hour of such interactive work, one can develop a strong sense of how outcomes vary with *combinations* of parameter values. This is much more than traditional sensitivity analysis. Moreover, one can search out and focus upon the "good" cases. Figure 1 is merely one schematic snapshot of the computer screen for choices of parameter values that show some successes. Most snapshots would be dominated by black squares because it is difficult to defend Kuwait against a large threat. Data View is not a commercial product, but RAND has made it available to government clients and some other organizations (e.g., allied military staffs).

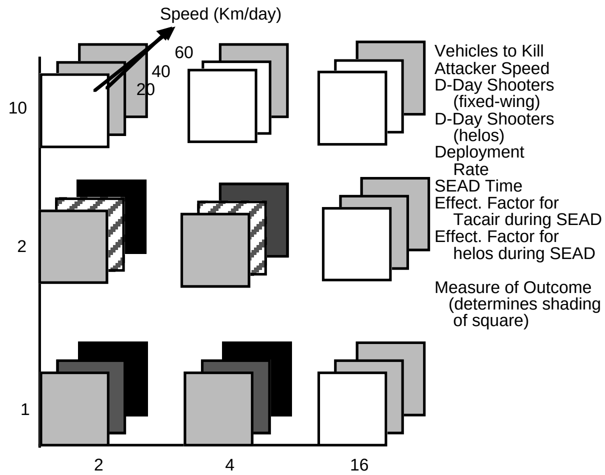

Figure 1: Display of Parametric Exploration

Other personal-computer tools can be used for the same purpose and the state of the art for such work is advancing rapidly. A much improved version of Data View called CAR™ is under development by Steve Bankes at Evolving Logic <www.evolvinglogic.com>. For those who prefer spreadsheet modeling, there are plug-in programs for Microsoft EXCEL® that provide statistical capabilities and some means for exploratory analysis. Two are Crystal Ball® <www.decisioneering.com> and @Risk® <www.palisade.com/.html/risk.html>. For a number of reasons such as visual modeling and convenient array mathematics, I usually prefer the Analytica® modeling system (the exception is when one needs procedural programming). Analytica <www.lumina.com> is an outgrowth of the Demos system developed at Carnegie Mellon University (Morgan and Henrion, 1992).

Figure 2 shows a screen image from recent work with Analytica on the same problem treated in Figure 1. In this case, we have a more traditional graphical display. Outcome is measured along the Y axis and one of the independent variables is plotted along the X axis. A second variable (D-Day shooters) is reflected in the family of curves. The other independent variables appear in the rotation boxes at the top. As with Data View, we change parameter values by clicking on a value and selecting from a menu of values. Such interactive displays allow us to "fly through the outcome space" for many independent parameters, in this case 9. For this number, the display was still quickly interactive for the given model and computer (a Macintosh PowerBook G3 with 256 MB of RAM).

So far, the examples have focused on parametric exploration. Figure 3 illustrates a hybrid exploration (Davis, et al. 1998). It shows the distribution of simulation outcomes resulting from having varied most parameter values "probabilistically" across an ensemble of possible wars, but with warning time and the delay in attacking armored columns left parametric.

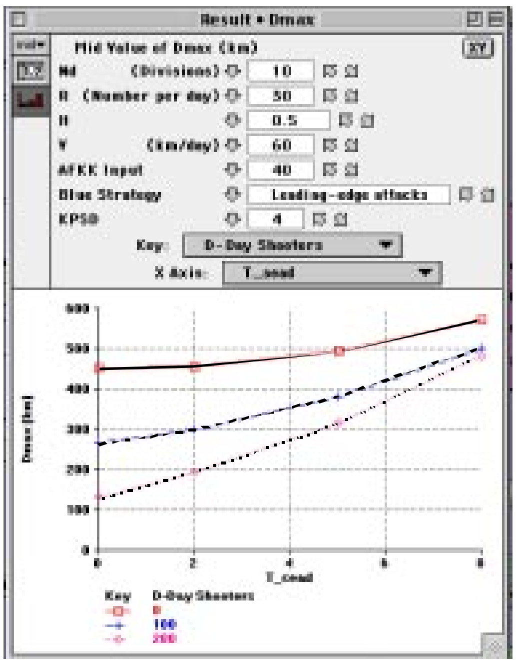

Figure 2: Analytica Display of Parametric Exploration

The probabilistic aspect of the calculation assumed, for example, that the enemy's movement rate had a triangular distribution across a particular range of values and that the suppression of air defenses would either be in the range of a few days or more like a week, depending on whether the enemy did or did not have air-defense systems and tactics that were not part of the best estimate. We represented this possibility with a discrete distribution for the likelihood of such surprises. The two curves in Figure 3 differ in that the one with crosses for markers assumes that interdiction of moving columns waits for suppression of air defenses (SEAD). The other curve assumes that interdiction begins immediately because the aircraft are assumed stealthy.

This depiction of the problem shows how widely the outcomes can vary and how the outcome distribution can be complex. The non-stealthy-aircraft case shows a spike at the right end where cases pile up because, in the simulation, the attacker halts at an objective of about 600km. Note that the mean is not a good metric: the "variance" is huge and the outcome may be multi-modal.

These results have been from analyses accomplished in recent years for the Department of Defense. As we look to the future, much more is possible with computational tools. Much better displays are possible for the same information and, even more exciting, computational tools can be used to aid in the search process of exploration. For example, instead of clicking through the regions of the outcome space, tools could automatically find portions of the space in which particular outcomes are found. One could then fine-tune one's insights by clicking around in that much more limited region of the outcome space. Or, if the model is itself driven by the exploration apparatus, then the apparatus could search for outcomes of interest and then focus exploration on those regions of the input space. That is, the experimental design could be an output of the search rather than an input of the analysis process. These methods are at the core of the evolving tool mentioned earlier called CAR (for Computer-Assisted Reasoning).

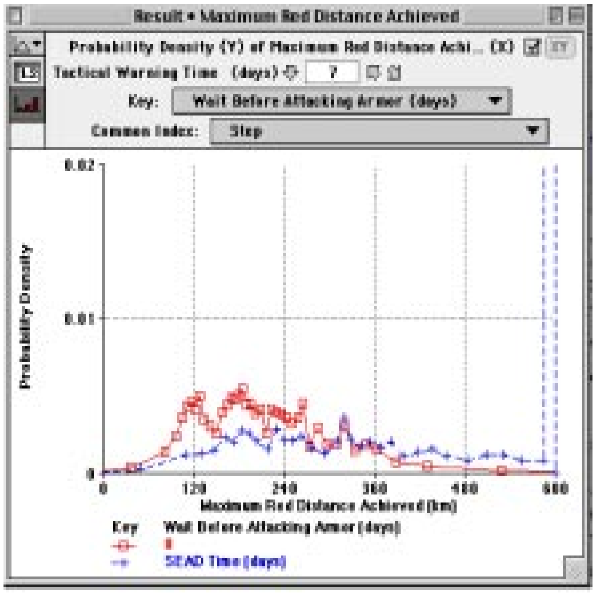

Figure 3: Analytica Display of "Probabilistic" Exploration

3 EXPLORATORY ANALYSIS IN CONTEXT

Exploratory analysis is an exciting development with a long history with RAND's RSAS and JICM models. How ever, it is only one part of a sound approach to analysis generally. It is worth pausing to emphasize this point. Figure 4 shows how different types of models and simulations (including human games) have distinct virtues. The figure is specialized to military applications, but a more generic version applies broadly to a wide class of analysis problems.

White rectangles indicate “good;” that is, if a cell of the matrix is white, then the type model indicated in the left column is very effective with respect to the attribute indicated in the cell’s column. In particular, analytical models (top left corner), which have low resolution, can be especially powerful with respect to their analytical agility and breadth. In contrast, they are very poor (black cells) with respect to recognizing or dealing with the richness of underlying phenomena, or with the consequences of both human decisions and behavior. In contrast, field experiments often have very high resolution (they may be using the real equipment and people), and may be good or very good for revealing phenomena and reflecting human issues. They are, however, unwieldy and inappropriate for studying issues in breadth. The small insets in some of the cells indicate that the value of the type model for the particular purpose can often be enhanced a notch or two if the models include sensible decision algorithms or knowledge-based models that might be in the form of expert systems or artificial-intelligence agents.

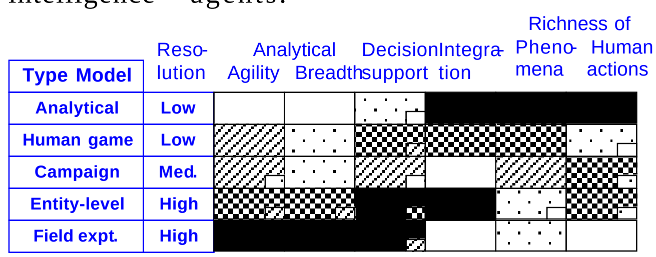

Figure 4: Virtues of a Model and Gaming Family

Figure 4 was developed as part of an exhortation to the Department of Defense regarding the need to have *families of models* and *families of analysis* (Davis, Bigelow, and McEver 1999). Unfortunately, government agencies often focus on a single model such as the venerable TACWAR, BRAWLER, or JANUS.

The niche of exploratory analysis is the top left hand corner of the matrix in Figure 4, which emphasizes analytical agility and *breadth* of analysis, rather than depth. However, the technique can be used hierarchically if one has a suitably modularized system model. One can do top-level exploration first and then zoom in. This is easier said than done, however, especially with traditional models. Specially designed models make things much easier, as discussed in what follows.

4 TECHNOLOGICAL ENABLERS

4.1 The Curse of Dimensionality

In principle, exploratory analysis can be accomplished with any model. In practice, it becomes difficult with large models. If F represents the model, it can be considered to be simply a complicated function of many variables. How can we run a computerized version of F to understand its character? If F has M inputs with uncertain values, then we could consider N values for each input, construct a full factorial design (or some subset, using an experimental design and sampling), run the cases, and thereby have a characterization. However, the number of such cases would grow rapidly (as N^M for full-factorial analysis), which quickly gets out of hand even with big computers. Quite aside from setup-and-run-time issues, comprehending and communicating the consequences becomes very difficult if M is large. Suppose someone asked "Under what conditions is F less than the danger point?" Given sufficiently powerful computers and enough time, we could create a data base of all the cases, after which we could respond to the question by spewing out lists of the cases in which F fell below the danger point. The list, however, might go on for thousands of pages. What would we do with the list? This is one manifestation of the curse of dimensionality.

4.2 The Need for Abstractions

It follows that, even if we have a perfect high-resolution model, we need abstractions to use it well. And, in the dominant case in which the high-resolution model is by no means perfect, we need abstractions that allow us to ponder the phenomena in meaningful ways, with relatively small numbers of cognitive chunks. People can reason with 3, 5, or 10 such cognitive chunks at a time, but not with hundreds. If the problem is truly complex, we must find ways to organize our reasoning. That is, we must decompose the problem by using principles of modularity and hierarchy. The need for an aspect of hierarchical organization is inescapable in most systems of interest—even though the system may be highly distributed and relatively nonhierarchical in an organizational sense.

A corollary of our need for abstractions is that *we need models that use the various abstractions as inputs.* It is not sufficient merely to display the abstracts as intermediate outputs (displays) of the ultimate detailed model. The reasons include the fact that when a decision maker asks a what-if question using abstractions, there is a 1:n mapping problem in translating his question into the inputs of a more detailed model. So also when one obtains macroscopic empirical information and tries to use it for calibration. Although analysts can trick the model by selecting a mapping, doing so can be cumbersome and treacherous. It is often better if the question can be answered by a model that accepts the abstractions as inputs.

4.3 Finding the Abstractions

Given the need for abstractions, how do we find them and how do we exploit them? Some guidelines are emerging (Davis and Bigelow 1998).

4.3.1 When Conceiving New Models and Families

With new models, the issue is how to *design*. Several options here are as follows:

- Design the models and model families top down so that significant abstractions are built in from the start, but do so with enough understanding of the microscopics so that the top-down design is valid.
- Design the models and families bottom up, but with enough top-down insight to assure good intermediate-level abstractions from the start.
- Do either or both of the above, but with designs taken from different perspectives.

The list does not include a pure top-down or pure bottom-up design approach. Only seldom will either generate a good design of a complex system. Note also the idea of alternative perspectives. For example, those in combat arms may conceive military problems differently than logisticians, and even more differently than historians attempting a macro-view explanation of events.

4.3.2 When Dealing With Existing Models

Only sometimes do we have the opportunity to design from scratch. More typically, we must adapt existing models. Moreover, the model "families" we may have to work with are often families more on the basis of assertion than lineage. What do we then do? Some possibilities here are:

- Study the model and the questions that users ask of the model to discover useful abstractions. For example, inputs X, Y, and Z may enter the computations only as the product XYZ. Or a decision maker may ask questions in terms of concepts like force ratio. For mature models, the displays that have been added over time provide insights into useful abstractions.
- Apply statistical machinery to search for useful abstractions. For example, such machinery might test to see whether the system's behavior correlates not just with X,Y, and Z, but with XY, XZ, YZ, or XYZ.
- Idealize the system mathematically and combine this with physical insight or empirical observation to guess at the form of aggregate behavior (e.g., inverse dependence on one variable, or exponential dependence on another). Consider approximations such as an integral being the product of the effective width of the integration interval and a representative non-zero value of the integrand.

The first approach is perhaps a natural activity for a smart modeler and programmer who begins to study an existing program, but only if he open-minded about the usefulness of higher-level depictions. The second approach is an extension of normal statistical analysis. The third approach is a hybrid that I typically prefer to the second. It uses one's understanding of phenomenology, and theories of system behavior, to gain insights about the likely or possible abstractions *before* cranking statistical machinery.

4.3.3 The Problem with Occam's Razor

The principle of Occam's razor requires that we prefer the simplest explanation and, thus, the simplest model. Enthusiasts of statistical approaches tend to interpret this to mean that one should minimize the number of variables. They tend to focus on data and to avoid adding variables for "explanation" if the variables are not needed to predict the data. In contrast, subject-area phenomenologists may prefer to enrich the depiction by adding variables that provide a better picture of cause-effect chains, but go well beyond what can be supported with meager experimental data. My own predilection is that of the phenomenologist, but with MRM designs one can sometimes have one's cake and eat it: one can test results empirically by focusing on the abstract versions of a model, while using richer versions for deeper explanation.

As an aside, a version of the Occam's Razor principle emphasizes use of the explanation that is simplest enough to explain all there is to explain, but nothing simpler! This should include phenomena that one "knows about" even if they are not clearly visible in the limited data. I would add to this the admonition made decades ago by MIT's Jay Forrester that to omit showing a variable explicitly may be equivalent to assuming its value is unity.

Competition among approaches can be useful. For example, phenomenologists working a problem may be convinced that a problem must be described with complex computer programs having hundreds or thousands of data elements. A statistical analysis may show that, despite the model's apparent richness, the system's resulting behavior is driven by something much simpler. This, in turn, may lead to a reconceptualizing of the problem phenomenologically. Many analogues exist in physics and engineering.

4.3.4 Connections Between New and Old Models

Although the discussion in Section 4.3.2 distinguished sharply between the case of new models and old ones, the reader may have noticed connections. In essence, working with existing models should often involve sketching what the models *should* be like and how models with different resolution *should* connect substantively. That is, working with existing models may require us to go back to design issues. Individuals differ, but I, at least, often find it easier to engage the problem than to engage someone's else's idiosyncratically described solution. Furthermore, I then have a better understanding of assumptions and approximations.

With this background, let me now turn to the design of multiresolution, multiperspective models and families (Davis and Bigelow 1999). Although this relates most directly to new models, it is relevant also to working with legacy models in preparing for exploratory analysis.

4.4 Multiresolution, Multiperspective Modeling

4.4.1 Definition

Multi-resolution modeling (MRM) is building a single model, a family of models, or both to describe the same phenomena at different levels of resolution, *and* to allow users to input parameters at those different levels depending on their needs. Variables at level n are abstractions of variables at level n+1. MRM is sometimes called variable-or selectable-resolution modeling. Figure 5 illustrates MRM schematically. It indicates that a higher level model (Model A) itself has more than one level of resolution. It can be used with either two or four inputs. However, in addition to its own MRM features, it has input variables that can either be specified directly or determined from the outputs of separate higher-resolution models (models B and C, shown as "on the side," for use when needed. In principle, one could attach models B and C in the software itself—creating a bigger model. However, in practice there are tradeoffs between doing that or keeping the more detailed models separate. For larger models and simulations, a combination single-model/family-of-models approach is desirable. This balances needs for analytical agility and complexity management.

MRM is not sufficient by itself because of the need for different abstractions or perspectives in different applications. That is, different perspectives—analogous to alternative representations in physics—are legitimate and important. They vary by conception of the system and choice of variables. Designing for both multiple resolution and multiple perspectives can be called MRMPM (pronounced Mr. MIPM).

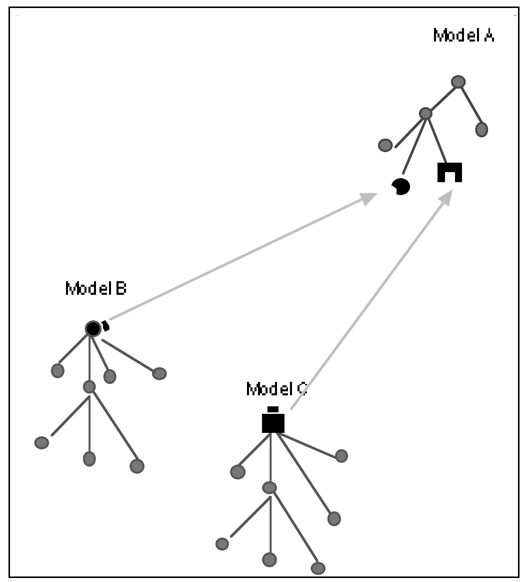

Figure 5: Figure 5: A Multiresolution Family

4.4.2 Mutual Calibration Within a Model Family

Given MRMPM models or families, we want to be able to reconcile the concepts and predictions among levels and perspectives. It is often assumed that the correct way to do this is to calibrate upward: treating the information of the most detailed model as correct and using it to calibrate the higher-level models. This is often appropriate, but the fact is that the more detailed models almost always have omissions and shortcomings. Further, different models of a family draw upon different sources of information—ranging from doctrine or even "lore" on one extreme to physical measurements on a test range at the other.

Figure 6 makes the point that members of a multiresolution model family should be *mutually* calibrated (National Research Council 1997). For example, we may use low-resolution historical attrition or movement rates to help calibrate more detailed models predicting attrition and movement. This is not straightforward and is often done crudely by applying an overall scaling factor (fudge factor), rather than correcting the more atomic features of the detailed model, but it is likely familiar to readers. On the other hand, much calibration is indeed upward. For example, a combat model with attrition coefficients should typically have adjustments of those coefficients for different circumstances identified in a more detailed model.

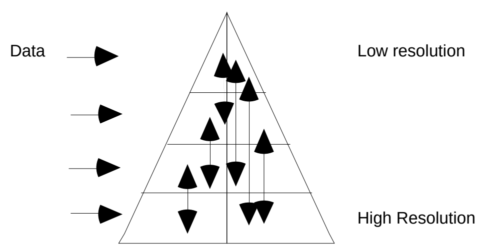

Figure 6: Mutual Calibration of Models in a Family

4.4.3 Design Considerations

So, given their desirability, how do we build a family of models? Or, given preexisting models, how do we sketch out how they "should" relate before connecting them as software or using them for mutual calibration? Some highlights are as follows.

The first design principle is to recognize that there are limits to how well lower-resolution models can be consistent with high-resolution models. *Approximation is a central concept from the outset.* Several points are especially important:

- Consistency between two models should be assessed in the context of use. What matters is not whether they generate the same final state of the system, but whether they generate approximately the same results in the application (e.g., rank ordering of alternatives). This ties into the well-known concept of experimental frames (Zeigler, et al. 2000).
- Consistency of aggregated and disaggregated models must also be judged recognizing that low-resolution models may reflect aggregate-level knowledge not contained in the detailed model.
- Comprehensive MRM is very difficult or impossible for complex M&S, but having even some MRM can be far more useful than having none at all.
- Members of an MRM family will typically be valid for only portions of the system's state space. Parameter values (and even functional forms) should change with region.
- Mechanisms are therefore needed to recognize different situations and shift models. In simulations, human intervention is one mechanism; agent-based modeling is another.
- Valid MRM will often require stochastic variables represented by probability distributions. Further, valid aggregate models must sometimes reflect correlations among variables that might naively be seen as probabilistically independent.

With these observations, the ideal for MRM is a hierarchical design for each MRM process, as indicated in Figure 5.

4.4.4 Desirable Design Attributes

From the considerations we have sketched above, it follows that models and analysis methodologies for exploratory analysis should have a number of characteristics. First, they should be able to reflect hierarchical decomposition through multiple levels of resolution and from alternative perspectives representing different "aspects" of a system.

Less obviously, they should also include realistic mechanisms for the natural entities of the system to act, react, adapt, mutate, and change. These mechanisms should reflect the relative "fitness" of the original and emerging entities for the environment in which they are operating. Many techniques are applicable here, including game-theoretic methods and others that may be relatively familiar to readers. However, the most fruitful new approaches are those typically associated with the term agent-based modeling. These include sub-models that act "as the agents for" political leaders and military commanders or—at the other extreme— infantry privates on the battlefield or drivers of automobiles on the highway. In practice, such models need not be exotic: they may correspond to some relatively simple heuristic decision rules or to some well-known (though perhaps complex) operations-research algorithm. But to have such decision models is quite different from depending on scripts.

Because it is implausible that closed computer models will be able to meet the above challenge in the foreseeable future, the family of "models" should allow for human interaction—whether in human-only seminar games, small-scale model-supported human gaming, or distributed interactive simulation. This runs against the grain of much common practice.

4.4.5 Stochastic Inputs To Higher Level Models

The last item in the above list is often ignored in today's day-to-day work. Indeed, too often models that need to be stochastic are deterministic, with quantitatively serious consequences (Lucas 2000). Often, workers calibrate a high-level (aggregate) model using average outcomes of allegedly "representative" high-resolution scenarios. For example, a theater-level model's air model might be calibrated to results of detailed air-to-air simulation with Brawler, which treats individual engagement classes (e.g., 1 on 1, 1 on 2, …4 on 8). This may appear to establish the validity of the theater-level model, but in fact the calibration is treacherous. After all, what kinds of engagements occur may be a sensitive function of the sides' command and control systems, strategies, and weather. The calibrations really need to be accomplished on a highly study-specific basis.

Furthermore, the higher-level model inputs often need to be stochastic. Figure 7 illustrates the concept schematically for a simple problem. Suppose that a process (e.g., one computing the losses to aircraft in air-to-air encounters) depends on X,Y, S, and W. But suppose that the outcome of ultimate interest involves many instances of that process with different values of S and W (e.g., different per-engagement numbers of Red and Blue aircraft). An abstraction of the model might depend only on X,Y, and Z (e.g., overall attrition might depend on only numbers of Red and Blue aircraft, their relative quality, and some command and control factor). If the abstraction shown is to be valid, the variable Z should be consistent with the higher-resolution results. However, if it does not depend explicitly on S and W, then there are "hidden variables" in the problem and Z may appear to be a random variable, in which case so also would the predicted outcome F be a random variable. One could ignore this randomness if the distribution were narrow enough, but it might not be.

In the past, such calibrations have been rare because analysts have lacked both theory and tools for doing things better. The "theory" part includes not having good descriptions of how the detailed model should relate to the simplified one. The tool part includes the problem of being able to define the set of runs that should be done (representing the integral of Figure 7) and then actually making those runs.

Ideally, such a calibration would be dynamic within a simulation. Moreover, it would be easy to adjust the calibration to represent different assumptions about command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR), as well as tactics. We are nowhere near that happy situation today,

$$F(X,Y,S,W) \approx \tilde{F}(X,Y,\tilde{Z})$$

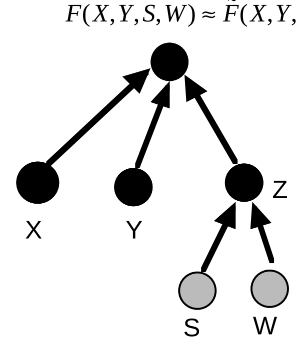

Figure 7: Input to Higher Level Model May Be Stochastic

5 RECENT EXPERIENCE AND CONCLUSIONS

MRMPM is not just idealized theory, but something usable. Over the last several years, my colleagues and I have done considerable work related to the problem of halting an invading army using precision fires from aircraft and missiles. The most recent aspects of that work included understanding in some detail how the effectiveness of such fires are affected by details of terrain, enemy maneuver tactics, certain aspects of command and control, and so on. This provided a good test bed for exploring numerous aspects of MRMPM theory (Davis, Bigelow, and McEver 2000).

For this work we developed a multiresolution personal-computer model (PEM), written in Analytica, to understand and extend to other circumstances the findings from entity-level simulation of ground maneuver and long-range precision fires. A major part of that work was learning how to inform and calibrate PEM to the entity-level work. There was no possibility, in this instance, of revising the entity-level model. Nor, in practice, did we have such a good understanding of the model as to allow us to construct a comprehensive calibration theory. Instead, we had to construct a new, more abstract, model and attempt to impose some of its abstractions on the data from runs of the entity-level simulation in prior work, plus some special runs made for our purposes. The result is a case history with what are probably some generic lessons learned.

Figure 8 illustrates one aspect of PEM's design. It shows the data flow within a PEM module that generates the impact time (relative to the ideal impact time) for a salvo of precision weapons aimed at a packet of armored fighting vehicles observed by surveillance assets at an earlier time. Other parts of PEM combine information about packet location versus time and salvo effectiveness for targets that happen to be within the salvo's "footprint" at the time of impact, to estimate effectiveness of precision weapons. For the salvo-impact-time module, Figure 8 shows how PEM is designed to accept inputs as detailed as whether there is enroute retargeting of weapons, the latency time, and weapon flight time. However, it can also accept more aggregate inputs such as time from last update. If the input variable Resolution of Time of Last Update Calculation is set "low," then Time From Last Update is specified directly as input; if not, it is calculated from the lower-level inputs.

This design has proven very useful—both for analysis itself and for communicating insights to decision makers in different communities ranging from the C4ISR community to the programming and analysis community. In particular, the work clarified how the technology-intensive work of the C4ISR acquisition community relates to higher-level strategy problems and analysis of such problems at the theater level.

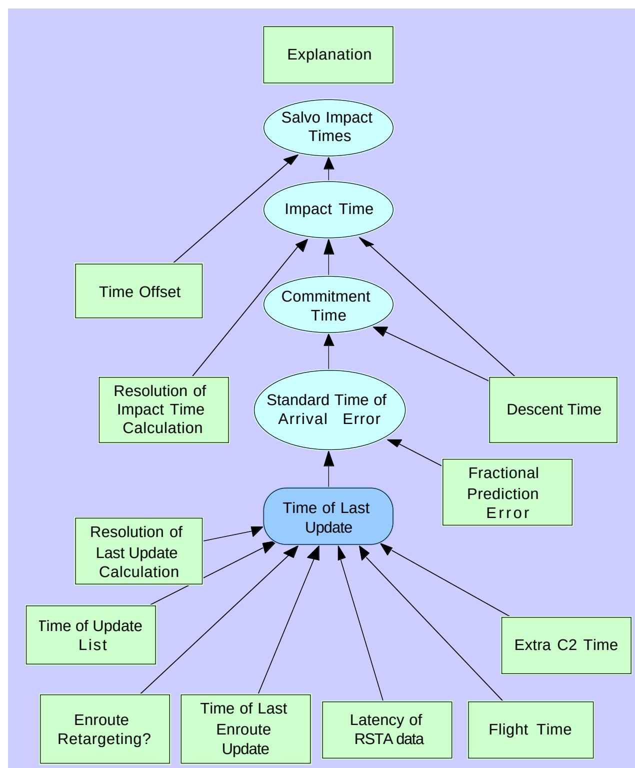

Figure 8: Multiresolution, Multiperspective Design

In other reports (McEver, Davis, and Bigelow, 2000a,b), we describe a broader but more abstract model (EXHALT) that we use for theater-level halt-problem analysis and experiments to deal with the multi-perspective problem. One conclusion is that MRMPM work rather demands a building-block approach that empasizes study-specific assembly of the precise model needed. Although we had some success in developing a closed MRMPM model with alternative user modes representing different demands for resolution and perspective (e.g., the switches in Figure 8), it proved impossible to do very much in that regard: the number of interesting user modes and resolution combinations simply precludes being able to wire in all the relevant user modes. Moreover, that explosion of complexity occurs very quickly. At-the-time-assembly from building blocks, not prior definition, is the stronger approach. This was as we expected, but even more so.

Fortunately, we were able to construct the models needed quickly—in hours rather than days or weeks—as the result of our building-block approach, visual modeling, use of array mathematics, and strong, modular, design.

We also concluded that current personal computer tools—as powerful as they are in comparison with those in past years—are not yet up to the challenge of making the building-block/assembly approach rigorous, understandable, controllable, and reproducible without unrealistically high levels of modeler/analyst discipline. Thus, there are good challenges ahead for the enabling–technology community. Also, the search models for advanced exploratory analysis are not yet well developed.

REFERENCES

Bankes, S. C. 1993. Exploratory modeling for policy analysis. *Operations Research* 41(3).

Bigelow, J. H., P. K. Davis, and J. McEver. 2000. Case history of using entity-level simulation as imperfect experimental data for informing and calibrating simpler analytical models for interdiction. In *Proceedings of the SPIE* Vol. 4026.

Davis, P. K (ed.). 1994. *New Challenges in Defense Planning: Rethinking How Much Is Enough*. RAND, Santa Monica, CA.

Davis, P. K. 2000. Multiresolution, multiperspective modeling as an enabler of exploratory analysis. In *Proceedings of the SPIE*, Vol. 4026.

Davis, P. K. and R. Hillestad. 2001. *Exploratory Analysis for Strategy Problems With Massive Uncertainty*. RAND, Santa Monica, CA.

Davis, P.K. and J. H. Bigelow. 1998. *Experiments in Multiresolution Modeling*. RAND, Santa Monica, CA.

Davis, P.K. and M. Carrillo. 1997. *Exploratory Analysis of the Halt Problem: A Briefing on Methods and Initial Insights.* DB-232, RAND, Santa Monica, CA.

Davis, P. K., D. Gompert, R. Hillestad, and S. Johnson. 1998. *Transforming the Force: Suggestions for DoD Strategy.* RAND Issue Paper, Santa Monica, CA.

Davis, P. K., D. Gompert,. and R. Kugler. 1996. *Adaptiveness in National Defense: the Basis of a New Framework.* RAND Issue Paper, Santa Monica, CA.

Davis, P.K., J. H. Bigelow, and J.McEver. 2000. *Effects of Terrain, Maneuver, Tactics, and C4ISR on the Effectiveness of Long Range Precision Fire.* RAND, Santa Monica, CA.

Davis, P.K., J. H. Bigelow, and J. McEver. 1999. *Analytic Methods for Studies and Experiments on Transforming the Force.* RAND, Santa Monica, CA.

Lempert, R., .M. E. Schlesinger, and S. C. Bankes. 1996. When we don't know the costs or the benefits: adaptive strategies for abating climate change. *Climatic Change* 33(2).

Lucas, Thomas. 2000. The stochastic versus deterministic argument for combat simulations: tales of when the average won't do. *Military Operations Research* 5(3).

McEver, J., P. K. Davis, and J. H. Bigelow. 2000a. *EXHALT: An Interdiction Model for the Halt Phase of Armored Invasions. RAND*, Santa Monica, CA.

McEver, J., P. K. Davis, and J.H. Bigelow. 2000b. Implementing multiresolution models and families of models: from entity level simulation to personal-computer stochastic models and simple 'repro models.' *Proceedings of the SPIE* Vol. 4026.

Morgan, G., and M. Henrion. 1992. *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis.* Cambridge, MA:Cambridge University Press.

National Research Council. 1997. *Modeling and Simulation: Technology for the United States Navy and Marine Corps, 2000–2035* Vol. 9. Washington, D.C:National Academy Press.

Zeigler, B. P.,T. G. Kim, and H. Praehofer. 2000. *Theory of Modeling and Design.* San Diego:Academic Press.

AUTHOR BIOGRAPHY

PAUL K. DAVIS is a senior scientist and Research Leader at RAND, and a Professor of Public Policy in the RAND Graduate School. He holds a B.S. from the U. of Michigan and a Ph.D. in Chemical Physics from the Massachusetts Institute of Technology. Dr. Davis is a recipient of the Wanner Memorial Award of the Military Operations Research Society. He is a member of the Naval Studies Board under the National Research Council and has served on a number of studies for the Council, the Defense Science Board, and the National Institute for Standards and Technology. His addresses are <pdavis@rand.org> and <www.rand.org/personal/pdavis>.

Informing and Calibrating A Multiresolution Exploratory Analysis Model With High Resolution Simulation: The Interdiction Problem As A Case History

James Bigelow, Paul K. Davis,
and Jimmie McEver

Reprinted from
A preliminary version was published in the *Proceedings of the SPIE,* Vol. 4026, p. 26-36, Enabling Technology for Simulation Science IV, Alex F. Sisti; Ed., June, 2000. The final version reprinted here was published (with Davis as conference presenter, shown as first author) in *Proceedings of the 2000 Winter Simulation Conference,* Jeffrey A. Joines, Russell R. Barton, K. Kang, and Paul A. Fishwick (editors), pp. 316-325.

National Defense Research Institute

INFORMING AND CALIBRATING A MULTIRESOLUTION EXPLORATORY ANALYSIS MODEL WITH HIGH RESOLUTION SIMULATION: THE INTERDICTION PROBLEM AS A CASE HISTORY

Paul K. Davis
James H. Bigelow
Jimmie McEver

RAND
1700 Main Street
Santa Monica, CA 90407-2138

ABSTRACT

Exploratory analysis uses a low-resolution model for broad survey work. High-resolution simulation can sometimes be used to inform development and calibration of such a model. This paper is a case history of such an effort. The problem at issue was characterizing the effectiveness, in interdicting an invading army, of long-range precision fires. After observing puzzling results from high-resolution simulation, we developed a multiresolution personal-computer model called PEM to explain the phenomena analytically. We then studied the simulation data in depth to assess, adjust, and calibrate PEM, while at the same time discovering and accounting for various shortcomings or subtleties of the high-resolution simulation and data. The resulting PEM model clarified results and allowed us to explore a wide range of additional circumstances. It credibly predicted changes in effectiveness over two orders of magnitude, depending on situational factors involving C4ISR, maneuver patterns, missile and weapon characteristics, and type of terrain. The insights gained appear valid and a simplified version of PEM could be used for scaling adjustments in comprehensive theater-level models.

1 INTRODUCTION

1.1 Background

An important operational challenge for the U.S. military is being able to halt an invading armored force *early*. Being able to do so is plausible, for some scenarios of considerable interest, if U.S. forces acquire the appropriate weapon systems, organization, and doctrine. Doing so would be part of the larger effort to "transform U.S. military forces" as called for in the Quadrennial Defense Review (Cohen 1997, see also Davis et al. 1998).

Meeting the challenge will not be easy and it is by no means clear as yet how successful ongoing developments and efforts to change organization and doctrine will be. Analysis can contribute to understanding what is needed and how the effectiveness of interdiction efforts would vary with scenario. This paper, based on a much more lengthy report (Davis, Bigelow, and McEver 2000), and building upon a preliminary presentation (Bigelow, Davis, and Bigelow 2000), is a case history of how one can use a family-of-models approach that includes multiresolution exploratory-analysis models and a pre-existing high-resolution simulation.

Reprinted by permission from *Proceedings of the 2000 Winter Simulation Conference*, Jeffrey A. Joines, Russell R. Barton, K. Kang, and Paul A. Fishwick, editors, pp. 316-325. (A preliminary version appeared in *Proceedings of SPIE*, Vol. 4026 pp. 26-36, "Enabling Technology for Simulation Science IV," Alex F. Sisti, editor, June 2000.)

1.2 The Analytical Problem

The idealized military problem here is one in which an invading armored force with many armored fighting vehicles (AFVs) pours across a border along one or a few major roads and moves rapidly toward an objective that may be some 100s of km from the border. An important issue is whether a defender could halt that invading army quickly by interdicting it with long-range precision fires in the form of aircraft with precision weapons such as JDAM or JSOW, and long-range missiles such as the Army ATACMS firing the "brilliant" munition BAT. The answer, of course, should depend upon the number of forces immediately present, deployment rates, weapon effectiveness, the size and speed of the advance, and so on. Clarifying those and other dependences is a natural task for exploratory analysis.

An exploratory analysis in 1997 highlighted some of the key issues for desert circumstances (Davis and Carrillo 1997), some of which were examined in considerable detail the next year (Ochmanek et al. 1998). It was evident, however, that mixed-terrain cases would be different—although it was widely believed that the differences would not be dramatic because roadways are often open. More detailed modeling was needed to understand the issues.

In fact, RAND has long used a detailed, entity-level force-on-force suite of models in defense studies. A number of these studies, led by Randall Steeb and John Matsumura, have considered long range precision fires, as well as upgraded ground forces, in scenarios dealing with defense against an armored invasion. In work done for the 1996 and 1998 Summer Studies, and for the Army, RAND found (Matsumura et al. 1997, Matsumura et al. 1999) very different effectiveness for long range fires of the ATACMS/BAT combination. In simulations for the DSB '96 summer study, ATACMS/BAT killed about three Red vehicles per missile, which was already lower than many individuals expected of this weapon. So it was a most unwelcome surprise when, in simulations for the DSB '98 summer study, the same weapon killed a factor of 5-10 fewer Red vehicles per missile.

There were some possible explanations. In DSB '96 the terrain was entirely open, while the DSB '98 terrain had a good deal of tree cover. In DSB '96 almost all the Red vehicles were armored fighting vehicles (AFVs), while in DSB '98 fewer than 20 percent of the Red vehicles were AFVs. And in DSB '96 the Red vehicles were in dense formations (50–100 meter spacing), while in DSB '98 vehicles were much more dispersed.

These arguments might seem to rationalize results, but counter arguments suggested caution. For example, in DSB '98 the missiles were aimed only at clearings, and only when the human-in-the-loop targeter (an Army officer trained in ATACMs doctrine) projected AFV arrivals based on simulated C4ISR information. One could therefore argue that tree cover should reduce the number of missiles launched, but not the effectiveness per missile. Also, the BAT submunition preferentially homes in on AFVs, so the presence of trucks should make little difference. And the large footprint of the ATACMS/BAT (a radius of at least four kilometers) should negate the large separations between vehicles. It seemed clear that we should not be satisfied with glib rationales, but should instead study the issues more carefully.

Doing so was not straightforward. Although the entity-level simulation is a rich description of phenomenology and has clear physics algorithms and a rooting in "hard" weapon data, the analytical implications for operational-level effectiveness are often difficult to understand because of the simulation's bottom-up character and huge number of variables. Further, using the simulation is manpower-intensive; it is not practical to consider a wide range of scenarios (although many variations can be made in, say, weapon Pks, detection probabilities, and entity characteristics).

To try to understand better the higher-level issues while connecting them with the microscopics, we built a personal computer model called the PGM Effectiveness Model (PEM) (Davis, Bigelow and McEver, 2000). This was an intermediate-level model in that, in *some* respects, it had low resolution in comparison with RAND's entity-level simulation suite, but higher resolution than models used for theater-level analysis. PEM was focused on a small part of the overall problem: effectiveness of long-range fires in interdicting a particular group of AFVs amidst terrain. PEM is not small enough to be written on the back of an envelope, but it is nonetheless quite small and simple. The conceptual core is based on simple physics. We implemented it in Analytica™, a very flexible visual modeling tool.

One can think of PEM as a stochastic, physically and mathematically-based, scaling model (not a mere statistical fit) that adjusts the effect of long range precision fires for the influence of a variety of factors. These factors include the time of last update, which operates through the error in the missile arrival time; the footprint of the weapon; the openness of the terrain; and the density of the Red formation. We have used the model to investigate interactions among the factors. We have also developed an even simpler deterministic version of the model that could be used as a subroutine to incorporate these factors in other models, such as EXHALT (McEver, Davis, and Bigelow 2000), RAND's JICM, or even DoD's emerging JWARS model.

1.3 Relating the Work To Generic Issues

Our work is an example of multi-resolution modeling (MRM) (Davis and Bigelow 1998), which is the practice of building mutually consistent models or families of models. PEM itself has multiple levels of resolution, which are related cleanly through hierarchical design. RAND's high-resolution simulation suite was developed years ago and is by no means integrated with PEM. However, we could do family-of-models work by investing the time necessary to understand relationships and accomplish some calibrations. Doing so would illustrate common difficulties in working with legacy models.

Multiresolution modeling is important for many reasons, the most fundamental of which is perhaps the need of humans to reason at different levels of detail. Such reasoning requires variables that can be manipulated (i.e., inputted) at those different levels to discuss cause and cause-effect relationships. This implies the need for models at different. Even excellent high-resolution models do not meet this need. Other reasons for MRM and its cousin multiresolution, multiperspective modeling (MRMPM) (Davis 2000) involve tradeoffs between agility and phenomenological richness, the need to connect to different levels of empirical data, costs, time, and the treatment of uncertainty. For example, we need low-resolution models for exploratory analysis, but we need high-resolution models to understand underlying phenomena, to provide links to physical entities and concrete low-level options, and, sometimes, to calibrate the lower-resolution models.

With this background, let us now proceed as follows. Section 2 describes the models we used for our analysis; Section 3 describes our analysis of high-resolution data; and Section 4 draws some lessons learned. Davis developed the PEM model; Bigelow did the extensive data analysis reported here; and McEver developed a simplified “repro model” version of PEM.

2 THE MODELS

2.1 RAND’s Force-On-Force Modeling Suite

The high resolution models that produced the provocative results motivating our study provide RAND with a valuable capability for high fidelity analysis of force-on-force encounters. In this suite, the RAND version of JANUS (a model originally developed by the Lawrence Livermore Laboratories) serves as the primary force-on-force combat effectiveness simulation and provides the overall battlefield context, modeling as many as 1500 individual systems on a side. The Seamless Model Interface (SEMINT) integrates JANUS with a host of other programs into one coordinated system, even though the participating models may be written in different programming languages, and run on different hardware under different operating systems. In effect, SEMINT gives RAND the ability to augment a JANUS simulation with specialized high fidelity models, without modifying the basic JANUS algorithms. The result is distributed and sometimes interactive simulation (DIS) for analysis, although the models in this case are all located in the same laboratory. The system was developed prior to the High Level Architecture (HLA) that is becoming the DoD standard geographically distributed work.

As currently configured, JANUS conducts the ground battle, calling on the RAND Target Acquisition Model (RTAM) to provide more accurate calculation of detection probabilities of special low observable vehicles. The Model to Assess Damage to Armor by Munitions (MADAM), developed originally by the Institute for Defense Analyses, simulates the effects of smart munitions, including such aspects as chaining logic, multiple hits, and unreliable submunitions, while the Acoustic Sensor Program (ASP) provides a detailed simulation of acoustic phenomenology for such systems as air-delivered acoustic sensors and wide-area munitions. Should the conflict involve helicopter or fixed wing operations, the flight planners BLUE MAX II (fixed wing) and CHAMP (helicopter) determine flight paths for the missions, flown against the actual JANUS threat, and RAND's Jamming and Radar Simulation (RJARS) conducts the defense against the aircraft, including detection, tracking, jamming and SAM operations. The Cartographic Analysis and Geographic Information System (CAGIS), developed originally at the Johns Hopkins University Applied Physics Laboratory, provides consistent geographic information to all the simulations, while SEMINT passes messages among the models, and maintains a Global Virtual Time to keep the models in synchronization.

For our purposes, the Model to Assess Damage to Armor by Munitions (MADAM) is key. RAND has upgraded MADAM so that it models the technologies associated with the following munitions:

- Seek And Destroy ARMor (SADARM)
- Sensor-Fused Weapons (SFW-Skeet)
- Damocles
- Low-Cost Anti-Armor Submunition (LOCAAS)
- Terminally-Guided Weapon/Projectile (TGW/TGP)
- Precision Guided Mortar Munition (PGMM) (Infra-Red (IR) & Millimeter Wave (MMW))
- Brilliant Anti-Tank (BAT)
- Wide Area Munitions (WAM)

The model simulates target seeking logic, false alarm rates, hulks, submunition reacquisition, shots, hits and kills, as well as bus, munition, and submunition reliability. For example, to estimate how many vehicles are killed by a BAT, MADAM simulates the separation of the bus from the launch vehicle, the separation of submunitions from the bus, several stages of acoustic seeking and deployment by the submunitions as they descend, an IR detection stage and a final shot/hit/kill event for each submunition. The outcome at each stage is determined, in part, by a random draw.

2.2 High Resolution Study of Long Range Fires

RAND has used this suite of models as follows to study long range precision fires (Matsumura et al. 1997, 1999). JANUS simulates the movement of each vehicle in a Red force across a terrain. The analyst scripts this movement by specifying the initial position and nominal velocity of each vehicle, as well as the path the vehicle will follow. For a road march, the path is a road. If Red is attacking a Blue position, the path will include off-road maneuver.

Periodically, say every five minutes of simulated time, a snapshot of Red vehicle positions is provided to a man-in-the-loop who decides the aim points and impact times of the long range precision weapons (sometimes it is possible to automate this function). Each snapshot provides incomplete information on the positions of Red vehicles. If a vehicle is obscured by foliage, there is a probability P_{TREE} of seeing it. If the vehicle is in the open, the probability P_{OPEN} of seeing it is larger. These probabilities can be adjusted to represent different qualities of C4ISR.

Based on the vehicles he sees, the man-in-the-loop selects aim points and impact times for the long range weapons. He will aim only at open areas, because we have assumed that a vehicle obscured by foliage or hidden in a town or city is not vulnerable. In addition, the time between the snapshot and the impact time of any salvo based on that snapshot must be at least as long as a specified latency period, which includes the time to collect the information in the snapshot, plus a decision time, plus the flight time of the weapon. In DSB '98, the man-in-the-loop would identify a group of vehicles to shoot at, estimate how long it would take to arrive at a nearby clearing, and lead the target as a duck hunter would lead his flying prey.
Finally, MADAM simulates the effect of the weapon on the Red vehicles near its aim point at its time of impact.

2.3 PGM Effectiveness Model (PEM) Concepts

Figure 1 illustrates the concepts on which PEM is built. PEM assumes that a column of Red vehicles is traveling along a road through a clearing of width W. Rather than being uniformly spaced, the Red vehicles are grouped into packets, perhaps representing platoons. Each packet has N AFVs separated from one another by a distance S. Successive packets (not shown) are separated by a distance P, which is larger than S. This column of vehicles moves through the clearing at a velocity V.

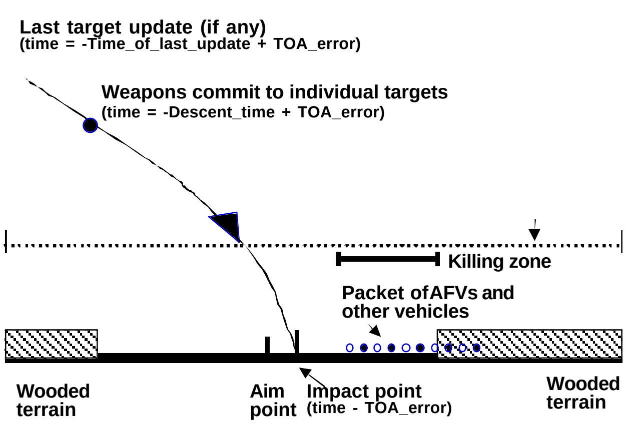

Figure 1: PEM Concepts

Blue attacks the column by firing a salvo of one or more missiles at the clearing, timed to arrive when a selected packet is expected to be in the center of the clearing. There is a random error in impact point and, more significant, a random error in the arrival time (TOA_error), the standard deviation of which is proportional to the time since the missile last received information about the projected position of the target packet (Time_of_last_update). If TOA_error is too large, the target packet may have passed completely through the clearing, or beyond the weapon's footprint F, whichever is larger; or (if the missile arrives early) the target packet may not have entered the clearing or the footprint. A smaller error will find the target packet not centered in the clearing, and part of it may be hidden in the trees (or urban clutter) on either end of the clearing.

Depending on the various parameters, parts of the packets just forward and rearward of the target packet may be in the killing zone. The presence or absence of these neighboring packets can change in kills per salvo by a factor of 2 or 3. The size of this effect depends on the separation between packets and the standard deviation of TOA error.

Except for a few details, this determines how many Red AFVs are in the killing zone of the weapon at its time of impact. PEM assumes the number of AFVs actually killed will be a specified fraction of the vehicles in the killing zone, up to a specified maximum. Actually, this was initially a hypothesis to be tested by comparing with the high-resolution data. As we shall see, truth is more complex.

3 CALIBRATING PEM TO JANUS/MADAM

A major part of our effort was data analysis—treating simulation data very much like experimental data (including recognition that the experimental conditions were sometimes not what they at first appeared to be, that the experiment [simulation model] was imperfect, that some of the data was flawed, and that not enough information was retained to determine all the causes and effects).

In what follows, we discuss four aspects of calibrating PEM to the high resolution models. First, we determine the lengths of clearings to use in PEM. In the high resolution model, this corresponds to selecting candidate aim points for the long range precision weapon. Next, we determine the Red order of march, i.e., the PEM parameters of vehicles per packet, vehicle and packet separations, and vehicle velocity. Third, we estimate the parameters that determine the missile arrival time error. In the high resolution model, these two aspects of calibration correspond to identifying a lucrative group of vehicles to target, and estimating when the group will arrive in a clearing. Fourth, we estimate weapon effectiveness.

3.1 Lengths of Clearings

A clearing is a basic PEM concept entirely characterized by its width. But no such concept exists in JANUS/MADAM: users of the model viewing displays of simulated behavior "see" clearings that are consequences of the microscopic terrain data bases, but they are visual abstractions, not something built in. One of our most interesting discoveries (something weapon engineers have undoubtedly come across over the years, but something to which we were previously insensitive) was that the definition of a clearing depends on point of view. The man-in-the-loop sees a snapshot of Red vehicle positions from the point of view of a long range reconnaissance device, perhaps a UAV or J-STARS orbiting a hundred kilometers or more from the target area. Its field of view must be wide enough to take in a large portion of the Red formation. Thus, the man-in-the-loop may miss small clearings, but may also fail to see that what appear to be unbroken open areas are in fact cluttered with small stands of trees or villages.

By contrast, when the weapon arrives over the aim point, it sees the local terrain at much higher resolution. What the man-in-the-loop thought was a clearing may be chopped up into very short stretches of open road (middle of Figure 2). What the man-in-the-loop thought was an unbroken stretch of trees may contain a long, open corridor (lower left of Figure 2). Such corridors along roads are common, though for different reasons, in both the real world and in the model.

Of course, whether vehicles in an open area are really vulnerable depends also on weapon characteristics, including their tolerance for “clutter” in the target area.

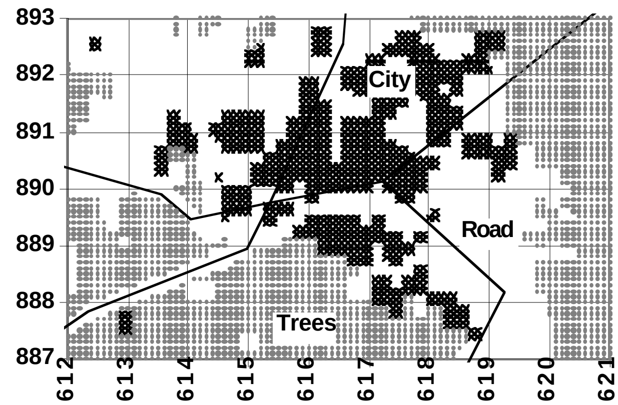

Figure 2: A Clearing at High Resolution

The ambiguity in how to define a clearing in the JANUS/MADAM simulations means that we cannot calibrate PEM simply by measuring open spaces in the high resolution representation of the terrain. PEM needs the distribution of open-area widths as seen by the arriving weapon—i.e., the distribution of small open-area widths (the small white areas in Figure 2). Since measuring these areas precisely is tedious, the distribution of widths considerable, and the precise distribution dependent on the particular physical area of battle, we calibrated PEM only approximately using a triangular distribution. The mode of that distribution for open-area width is a variable parameter in exploratory analysis.

3.2 Red Order of March

In PEM, the Red order of march is specified by four parameters: the number of AFVs in a packet, the distance between successive vehicles within a packet, the separation between packets, and the speed of vehicles across clearings. In any particular PEM run, all these parameters are constant, though we investigate their influence by changing them stochastically or parametrically from one run to another. Thus PEM assumes the Red formation is highly regular, described by only a few parameters. In the high resolution cases, the Red formations are rather irregular, so we must approximate.

In the DSB '98 study, the Red columns deviated from this simple description in several ways. First, there were three types of packets. One consisted of AFVs, the other two of scout vehicles and (mostly) trucks, respectively. In total, only 104 of the 543 Red vehicles in the simulation were AFVs. Second, not all AFV packets had the same numbers of vehicles separated by the same distances. AFV packets had from three to ten vehicles (average 6.7), and were separated by 150 to 600 (average 350) meters. And successive packets of AFVs were separated by from 1000 to 3800 meters. The speed of AFVs, however, was nearly constant at 76 kilometers per hour. This was a deliberately stressful case for long-range fires, much more so than in most previous RAND work.

The three kinds of packets in the DSB '98 summer study moved at different speeds. Reconnaissance, AFV, and truck packets began the simulation in overlapped positions, but as the simulation progressed the reconnaissance packets pulled ahead and the truck packets fell behind. All packets of the same kind moved forward in lockstep at a nearly constant speed.

In the DSB '96 summer study, the Red order of march was quite different. First, virtually all vehicles (458 out of 504) were AFVs. Second, they did not move in column formation for the whole simulation. Rather, they moved in columns for roughly the first 30 minutes of the simulation, and then redeployed for an attack on a Blue position. As the Red force redeployed into attack formation, vehicles were still densely packed. However, since they were no longer following the roads, it became harder to predict where they were going, and hence to lead them with long range fires. On the other hand, once the Red vehicles left the roads, they slowed. About two-thirds of the salvos were aimed at Red vehicles in column formation and the remainder at vehicles in attack formation.

While they were in column formation, the AFVs were packed much more densely than in the DSB '98 cases. Separations of 50 to 100 meters between AFVs were typical (even these separations, of course, are higher than in most historical battles). The groups of AFVs we might identify as packets for PEM often contained 50 vehicles or more.

3.3 Error in Weapon Arrival Time (TOA_error)

In PEM, the error in the weapon arrival time is the difference between the time the weapon arrives and the time the target packet is centered in the clearing. A negative TOA_error indicates that the weapon has arrived early, and a positive error that it has arrived late. We have assumed that the error is random with nearly zero mean and a standard deviation that is proportional to the time of last update.

The time of last update is the last time the shooter has the opportunity to adjust the aim point or impact time of the weapon. In the DSB '98 study, the man-in-the-loop had to specify the aim point and impact time of each ATACMS based on information that would be, for some cases at least 11 minutes old, and for other cases as much as 20 minutes old, at the time of impact.

The standard deviation of the TOA_error, as mentioned earlier, is proportional to the time of last update; and the constant of proportionality is the fractional error in the shooter's estimate of the speed of the AFVs along the road or track. The fractional speed error must have been considerable in the DSB '98 cases, since most ATACMS missile salvos found few AFVs in their footprints when they arrived over their aim points. Some of this was due to the fact that the shooter misguessed the route a group of Red vehicles would take, i.e., thought they would turn right instead of left. But most missiles fell on the routes that the Red columns actually did follow. Note, that the error in movement rate would not be remedied by a more accurate radar measuring instantaneous velocity, because the issue is estimating future speed along a curved and sometimes complex road.

We could not estimate the fractional-speed or time-of-arrival error directly, because the man-in-the-loop did not keep records on which group of vehicles he was targeting with each shot. But we could do it indirectly. Imagine that a camera floats above the aim point of a particular ATACMS salvo, and counts the number of vehicles in the ATACMS footprint (a circle with a nominal ATACMS kill radius). If we plot that number as a function of time, we obtain a figure such as Figure 3. The upper curve is the count of all vehicles; the lower curve is of AFVs only.

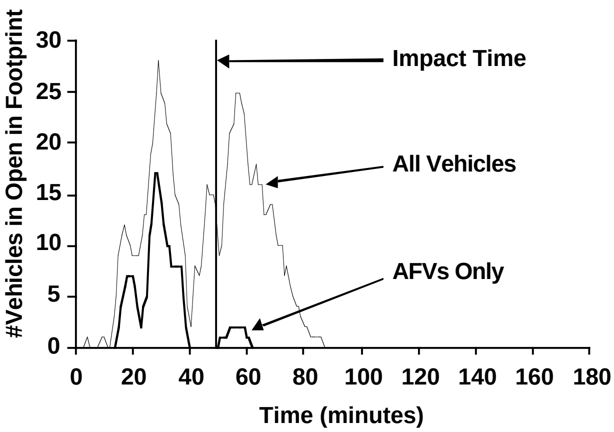

Figure 3: Vehicles in a Typical DSB '98 Footprint

Two things stand out. First, the number of AFVs in the footprint is highly variable and even small deviations from the optimal impact time will reduce the number of targets substantially. Second, there are frequently many vehicles in the footprint but no AFVs. Thus, if the shooter can't distinguish AFVs from other vehicles, he will surely waste a substantial number of shots.

Following up on this last point, we can define two windows of opportunity (WOP). The first is the interval of time from the entry of the first vehicle of any kind into the footprint, to the exit of the last vehicle from the footprint. The second is the time interval similarly defined for AFVs rather than for all vehicles.

Table 1 evaluates the timing of salvos in three DSB '98 cases, with times of last update of 20, 15, and 11 minutes, respectively. In all three cases, virtually all shots fell within the window of opportunities for all vehicles, suggesting that the shooter could assess when some part of the whole column would be in the footprint, even if he could not do so for a specific packet.

The shooter was less successful at estimating when the AFVs in the column would pass through the footprint, though he did better than chance: In each case, the window of opportunity for AFVs was about three-eighths of the window of opportunity for all vehicles, but the shooter was able to put more than half of his shots into the AFV window. The reason for this was probably not due to the shooter's ability to distinguish one vehicle type from another, although in principle he had the information to do so. Rather, the shooter had a "template" of the Red column that placed scout vehicles in front, followed by combat vehicles. The template placed support vehicles at the rear. The shooter targeted the portions of the column that his template suggested would contain the combat vehicles.

Table 1: Measures of Success in Timing of Shots

	Cases		
	X	Y	Z
Delay	20	15	11
	Percent of shots		
In WOP for all vehicles	100	94	100
In WOP for AFVs only	65	56	71
100*WOP(AFV)/WOP (All)	38	39	37
	Avg. AFVs in footprint		
Actual shots	1.85	1.44	3.42
In WOP for all vehicles	1.49	1.61	1.40
In WOP for AFVs only	3.88	4.03	3.73

Next we compare the number of AFVs in the footprints in the actual shots with the average numbers in the two windows of opportunity. In cases X and Z, the actual number is between the average numbers in the two windows, and it would be in case

Y as well, if we eliminated the two shots that missed the larger window altogether. It is plausible to argue, therefore, that the shooter was unable to time his shots well enough to hit individual packets. He could have done as well simply to establish a window somewhere between the all-vehicle and AFV windows we have defined, and picked an impact time at random within his window.

Finally, then, we have an indirect way to calibrate PEM's TOA_error distribution to the high resolution results. From the JANUS/MADAM results we may take the variation over time in the number of AFVs in a footprint (as in Fig. 3), and turn it into a frequency distribution—i.e., we can determine the fraction of time in some window of opportunity that there are zero AFVs in the footprint, or one AFV, or any other number of AFVs. In PEM there is a probability distribution of AFVs in the footprint, and the variation in this distribution is affected by the standard deviation of TOA_error. We need only set the standard deviation so that the PEM probability distribution looks similar to the JANUS/MADAM frequency distribution.

If we place a camera above a typical aim point in the DSB '96 cases, we see a very different picture. In particular, the number of AFVs in the footprint is large (Figure 4) compared with the numbers we saw in the DSB '98 chart. As we shall see shortly, when the number of AFVs in the footprint is large enough, kills per ATACMS/BAT missile reaches a maximum, after which a further increase in the number of AFVs has no effect. For this footprint there is a window of 25 or 30 minutes within which a missile impact should achieve that maximum number of kills. Thus, precise timing is unnecessary against Red formations as dense as those in the DSB '96 study.

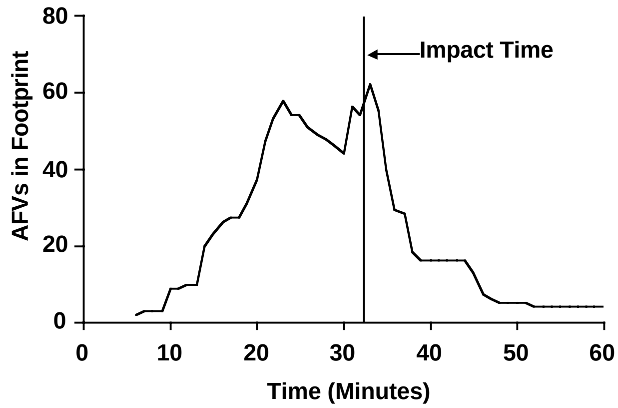

Figure 4: AFVs in a Typical DSB '96 Footprint

3.4 AFVs Killed per Salvo

In PEM, we calculate the number of AFVs killed per salvo as a function of the number of AFVs in the open in the salvo's footprint at the time of impact. The function we used initially was simple. The number of AFVs killed is proportional to the number in the footprint, up to a maximum (reflecting the finite number of submunitions in a BAT). As a result of the analysis below, we ended up adding a stochastic factor.

When we compare the initial, simple representation with actual salvos simulated in the DSB '98 and DSB '96 cases, inevitably we see errors. Part of the error is due to the fact that MADAM performs Monte Carlo trials to estimate kills for a given shot, and PEM's original relationship was deterministic. Another part is due to the fact that several types of vehicles are vulnerable to ATACMS/BAT, but they are not all equally vulnerable. In the high resolution simulations there are also such factors as background noise and dead vehicles that can interfere with the performance of the BAT submunition. Indeed, it is because of such terrain-and-case-dependent factors that the numbers we present here should be considered illustrative and unclassified. It is also why we consider PEM rather more of a scaling model than as a self-contained complete model. It cannot fully substitute for higher-resolution work.

One of the more interesting sources of error lies in the fact that BAT has differing effects against vehicles laid out in different patterns. ATACMS/BAT is particularly effective against a linear pattern of vehicles. Depending on the search algorithm assumed, it may be much less successful against a pair of crossed lines of vehicles. The crossed pattern confused the baseline search algorithm assumed for the BAT submunitions, and most of them fell harmlessly between the two lines. We emphasize, however, that alternative search algorithms exist and can also be used. The point here is that such details can matter a good deal. Such factors can only be understood with high-resolution work, not something as simple as PEM.

Whatever the reason, the number of AFVs killed per salvo is highly variable, even when one controls for the number of AFVs in the footprint. Figure 5 shows the variation across all the hundreds of simulated shots from the DSB ’98 study that had exactly five AFVs in the footprint.

Figure 6 plots the number of AFVs killed per salvo versus the number of AFVs in the footprint. We have included data from both the DSB ’98 and DSB ’96 studies. The points from the DSB ’98 study have a maximum of 14 AFVs per footprint. Each point represents the average kills from all shots with the same number of AFVs in their footprints. Points from the DSB ’96 study include salvos with very large numbers of AFVs in their footprints, and each point represents a single salvo of two ATACMS/BAT missiles. The solid line represents a plausible relation to use in PEM for calculating AFVs killed from AFVs in the killing zone. We fit it by eyeball, not by statistical methods, and ignored some points we had reason to believe were artifacts.

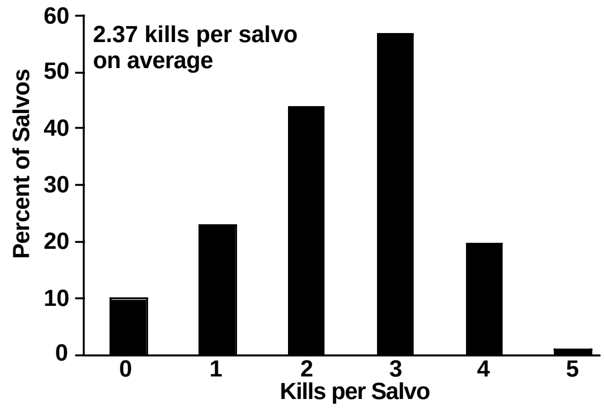

Figure 5: Variation in Kills for Salvos with Five AFVs in the Footprint

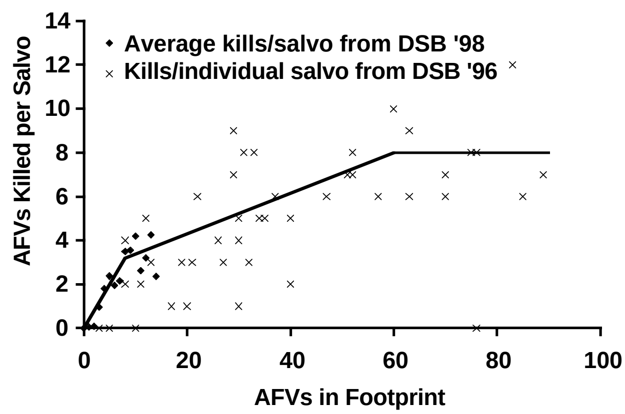

Figure 6: Kills per Salvo vs. AFVs in Footprin

Up to about eight AFVs in the footprint, the average number of AFVs killed per salvo is approximately 40 percent of the number of AFVs in the footprint. This portion of the data is dominated by results from the DSB ’98 study. When there are more than eight AFVs in the footprint, the trend line begins to flatten out. We knew it had to do so, since a salvo of two ATACMS dispenses a finite number of BAT submunitions, each of which is capable of a maximum of one kill. This portion of the data consists entirely of results from the DSB ’96 study.

In the current version of PEM, the relationship between AFVs in the footprint and AFVs killed is treated deterministically.

This discussion, however, suggests that it would be better to introduce a stochastic component if PEM is used in a context where statistical variation would be of interest. Otherwise, adding another stochastic component (as distinct from using distributions to reflect uncertainty in parameter values) is unnecessary.

4 LESSONS LEARNED

Many lessons could be learned from this work viewed as a case study. Some of the lessons were military; others were methodological.

4.1 Lessons About Long Range Precision Fires

In a sense, the reason each TACMS/BAT killed more AFVs in the DSB '96 study than in the DSB '98 study is simply that there turned out to be more AFVs in the weapons' footprints. However, building and exercising PEM has taught us *why*: what factors influenced the numbers of AFVs in the footprint, and how they interacted. The substantive lessons were reported in (Davis, Bigelow, and McEver 1999) and (Defense Science Board 1998).

Most of these results are reported elsewhere (Davis, Bigelow & McEver 2000). Briefly, the effectiveness of such fires can vary by two orders of magnitude, depending on the time of last update (which operates through the error in the weapon's time of arrival), the footprint of the weapon, the openness of the terrain, and the density of the Red formation. Moreover, these factors interact. Some examples: (1) if one shoots at small clearings, it becomes unimportant to use a weapon with a large footprint; (2) against low density Red formations it is vital that the weapon arrive at just the proper time; (3) against high density formations the TOA_error isn't important; and (4) if the weapon can loiter, TOA_error will also be less important.

The interaction of these factors affects one's choice of weapons. ATACMS/BAT has a large footprint but also a large TOA_error, while aircraft-delivered sensor fused weapons (SFW) have a small footprint but a low TOA_error. The ATACMS/BAT has an advantage over the SFW against high density Red formations in open terrain (large clearing), but loses its advantage if either Red's density is low or clearings are small.

In retrospect, none of these conclusions is counterintuitive, but before-the-fact intuition had been poor and PEM allowed us to crystallize a better intuition and develop quantitative relationships. Thus, we can estimate just how small the clearing must be, or how low the density of the Red vehicles, before the relative effectiveness of ATACMS/BAT versus SFW falls to any specified threshold. This would permit cost-benefit judgements about when to switch from one weapon to the other, or what weapon mix to use.

4.2 Lessons About Model Families

The exercise of building and calibrating PEM has provided lessons for the practice of building multi-resolution model families that involve a high-resolution legacy model that cannot readily be integrated with the family's higher level models. First, in any model at any level of resolution, variation of an output quantity from one case to another will be explained in part by variations in input parameters, and in part as the result of random events. Because a high resolution model will have more input parameters than a companion low resolution model, there is more scope to explain output variations by variations in input parameters. To achieve the same degree of variation in the low resolution model, it will often be necessary to introduce a random process. Parameters that were available to explain this part of the variation in the high resolution model, but which are missing in the low resolution model, are called hidden variables.

An example of this is the determination of the number of AFVs killed per salvo. In PEM, this is a function only of the number of AFVs in the footprint, and any variation from this function must be represented by a random process. In the high resolution model MADAM, it is a complicated function of the positions and types of all the vehicles—AFVs or not—in the neighborhood of the aim point, and the background noise generated by vehicles not very near the aim point. By changing the types and positions of vehicles for MADAM, we can produce variations in the number of AFVs killed per salvo, without changing the number of AFVs in PEM's footprint.

A second lesson is that concepts that seem well defined at one level of resolution may be ambiguous at another. The notion of a clearing is well defined in PEM. It has a definite width, and there is nothing more to know about it. But in the high resolution model, clearings must be identified from the description of the terrain and the roads. Is a stretch of road a clearing if trees line the road closely but the road itself is clear? Is it a clearing if it is interrupted by a few very short stands of trees? Must the treeless area be sufficiently large to be identified by a reconnaissance platform from miles away, or only large enough so trees don't interfere with the terminal search algorithm of a submunition?

Third, a high resolution model and its companion low resolution model may differ not only in their level of detail, but in their scope as well. Indeed, limiting the scope of a model as well as its detail is a way of keeping it small and agile. We built PEM to investigate long range precision fires against moving armored columns, and thus omitted all other vehicle types. The JANUS/MADAM suite of models includes the other vehicle types (e.g., trucks), and as we have seen, their presence affected the selection of impact times by the man-in-the-loop. He had to try to distinguish AFVs from other types of vehicles, and he did so with imperfect success.

ACKNOWLEDGMENTS

The authors benefited from collaboration and discussions with colleagues Randy Steeb, John Matsumura, and Tom Herbert, who lead RAND's high-resolution simulation work. The research reported here was sponsored by the Office of the Secretary of Defense.

REFERENCES

Bigelow, J.H., P.K. Davis, and J. McEver. 2000. Calibrating a policy-level exploratory analysis model with high-resolution simulation data for the interdiction problem. *Proceedings of the SPIE*. Vol. 4026.

Cohen, W.. 2000. *Report from the Quadrennial Defense Review*. Department of Defense, Washington, D.C.

Davis, P.K. 2000. Exploratory analysis enabled by multiresolultion, multiperspective modeling. In *Proceedings of the 2000 Winter Simulation Conference*.

Davis, P. K. and J. H. Bigelow. 1998. *Experiments in Multiresolution Modeling (MRM)*. MR-1004-DARPA, The RAND Corporation, Santa Monica, CA.

Davis, P. K., J. H. Bigelow, and J. McEver. 2000. *Effects of Terrain, Maneuver Tactics, and C4ISR on the Effectiveness of Long Range Precision Fires: A Stochastic Multi-Resolution Model (PEM) Calibrated to a High Resolution Simulation*. MR-1138-OSD, The RAND Corporation, Santa Monica, CA.

Davis, P.K., J.H. Bigelow, and J.McEver. 1999. *Analtyical Methods for Studies and Experiments on Transforming the Force.* CB-278-OSD, The RAND Corporation, Santa Monica, CA.

Davis, P.K. and M. Carrillo. 1997. *Exploratory Analysis of the Halt Problem: a Briefing on Methods and Initial Insights*. DB-232, The RAND Corporation, Santa Monica, CA.

Davis, P.K., D. Gompert, R. Hillestad, and S. Johnson. 1998. *Transforming the Force: Suggestions for DoD Strategy*. RAND Issue Paper, Santa Monica, CA.

Davis, P. K. and R. Hillestad. 2000. *Exploratory Analysis for Strategy Problems with Massive Uncertainty.* Draft in review. The RAND Corporation, January 23.

Defense Science Board. 1998. *1998 Summer Study Task Force on Operations Superiority in the 21st Century*. Office of the Under Secretary of Defense for Acquisition, Department of Defense, Washington, D.C.

Matsumura, J., R. Steeb, T. Herbert, M. Lees, S. Eisenhard, and J. Gordon. 1997. *Analytic Support to the Defense Science Board. Tactics and Technology for 21st*

Century Military Superiority. DB-198-A, The RAND Corporation, Santa Monica, CA.

Matsumura, J., R. Steeb, E. Isensee, T. Herbert, S. Eisenhard, and J. Gordon. 1999. *Joint Operations Superiority in the 21st Century*. Analytic Support to the 1998 Defense Science Board. DB-260-A/OSD, The RAND Corporation, Santa Monica, CA.

McEver, J., P.K. Davis, and J.A. Bigelow. 2000. *EXHALT: an Interdiction Model for the Halt Phase of Armored Invasions.* MR-1137, *The RAND Corporation*, Santa *Monica*, CA.

Ochmanek, D., Harshberger, E., T. Thaler, and G. Kent. 1998. *To find , and Not To Yield: How Advances in Information and Firepower Can Transform Theater Warfare*. MR-958, The RAND Corporation, Santa Monica, CA.

AUTHOR BIOGRAPHIES

PAUL K. DAVIS is a senior scientist and Research Leader at RAND, and a Professor of Public Policy in the RAND Graduate School. He holds a B.S. from the U. of Michigan and a Ph.D. in Chemical Physics from the Massachusetts Institute of Technology. Dr. Davis is a recipient of the Wanner Memorial Award of the Military Operations Research Society. He is a member of the Naval Studies Board under the National Research Council and has served on a number of studies for the Council, the Defense Science Board, and the National Institute for Standards and Technology. His addresses are <pdavis@rand.org> and <www.rand.org/personal/pdavis>.

JAMES H. BIGELOW is a senior mathematician at RAND. He holds a B.S. from the University of California at Berkeley and a Ph.D. in Operations Research from Stanford University. His research has dealt with medical, environmental, and transportation problems, as well as military problems such as logistics for the Army and Air Force, total force structure costs for all services, and total-Army design. His address is <James_Bigelow@rand.org>.

JIMMIE McEVER is an associate physical scientist at RAND. He holds a B.S. and Ph.D. in physics from the Georgia Institute of Technology and a Masters in Public Policy from Harvard's Kennedy School. Dr. McEver's current research involves applications of information technology in the Army, information security, science and technology issues in organizational settings, and alternative Air Force strategies for regional defense problems. His address is <Jimmie_McEver@rand.org>.

Implementing Multiresolution Models and Families of Models: From Entity-Level Simulation to Desktop Stochastic Models and "Repro" Models

Jimmie McEver, Paul K. Davis,
and James H. Bigelow

Reprinted from *Proceedings of the SPIE,*
Vol. 4026, p. 16-25, Enabling Technology for Simulation Science IV, Alex F. Sisti; Ed., June, 2000

National Defense Research Institute

Implementing multiresolution models and families of models: from entity-level simulation to desktop stochastic models and "repro" models

Jimmie McEver, Paul K. Davis and James Bigelow

RAND, 1700 Main St., Santa Monica, CA 90407-2138*

ABSTRACT

We have developed and used families of multiresolution and multiple-perspective models (MRM and MRMPM), both in our substantive analytic work for the Department of Defense and to learn more about how such models can be designed and implemented. This paper is a brief case history of our experience with a particular family of models addressing the use of precision fires in interdicting and halting an invading army. Our models were implemented as closed-form analytic solutions, in spreadsheets, and in the more sophisticated Analytica® environment. We also drew on an entity-level simulation for data. The paper reviews the importance of certain key attributes of development environments (visual modeling, interactive languages, friendly use of array mathematics, facilities for experimental design and configuration control, statistical analysis tools, graphical visualization tools, interactive post-processing, and relational database tools). These can go a long way towards facilitating MRMPM work, but many of these attributes are not yet widely available (or available at all) in commercial model-development tools—especially for use with personal computers. We conclude with some lessons learned from our experience.

Keywords: model abstraction, multiresolution models, variable resolution models, simulation, modeling environments, spreadsheet models, Analytica models.

1. INTRODUCTION

Other publications discuss "exploratory analysis" and the need for models with multiresolution, or both multiresolution and multiperspective features (MRM and MRMPM, respectively).[1,2,3] Briefly, MRM and MRMPM (pronounced "Mr. MiPM") refer to constructing and using models and families of models that are mutually consistent across multiple levels of resolution, and from multiple perspectives (or, in physics-like language, representations).

A key to building such models is constructing the proper hierarchical trees (including alternative branches) for the processes of interest. The trees should be functional and

---

*Correspondence: Email: Jimmie_McEver@rand.org; Telephone: 412 683 2300 x4910.

logical for modelers and users who move up and down tree branches (i.e., change resolution) or switch among alternatives (i.e., change perspectives).

This paper reviews some of our experiences in developing MRMPM tools for real analyses. In particular, we describe conclusions about the modeling environment needed to design and implement MRMPM, and about the way in which MRMPM families evolve. We end with some lessons learned about how to proceed.

2. ELEMENTS OF AN MRM/MRMPM MODELING AND ANALYSIS ENVIRONMENT

MRMPM work is in some respects fundamentally *different* than building a single, stand-alone model. As a result, different attributes are needed in the modeling environment if workers are to succeed rather than throw their hands up in frustration. These attributes are: [1]

- **Visual modeling**. Modeling environments should allow users to see data-flow level diagrams (and also object diagrams and process diagrams) of how processes and objects interact with each other. What is needed is not a collection of after-the-fact diagrams that the modeler must construct, but rather a set of dynamic displays that is generated by the environment as the builder works. Aside from the fact that after-the-fact documentation is often not done at all, after-the-fact diagramming is time-consuming and fraught with error. The point, after all, is to show the modeler what he has done, not what he intended to do. Visual modeling environments improve the design, accessibility and evolution of models built within them.

- **Interactive languages**. The ability to easily experiment with representations, processes, and other model-configuration parameters without having to recode and recompile is important for refining and iterating models, as well as for tailoring them to specific needs for resolution or perspective. Interactive languages are now common (e.g., in spreadsheets and ordinary languages), but when combined with a visual modeling environment, interactive languages can be especially clear and powerful enablers of collaboration and model evolution.

- **Facilities for experimental design and configuration control**. Modeling tools that allow for the construction of skeletal hierarchies and facilitate a thoughtful design phase before coding begins greatly improve quality and usefulness of the models that emerge. Tools for configuration control are important for enabling model evolution and tailoring for specific problems without creating “rogue variants” with only an uncertain relationship to base versions.

- **Statistical analysis tools**. Online statistical tools enable analysis of results, as well as mid-development checks that may be needed to develop and assess approximations (critical in MRMPM) or to estimate data and process validity.

- **Graphical visualization tools**. When analyzing model output data in a many-dimensional space, these tools can be key to generating insight into the important processes in the scenario space (i.e., the space spanned by all values of the model’s inputs).

- **Interactive post-processing and search**. It has often been asserted that the primary goal of modeling and simulation should be the generation of insight, rather than the provision of the “answer.” The search for such insights is greatly facilitated by interactive post-processing. Advanced search methods are also needed.[1]

- **Relational database tools**. When working with different levels of resolution and from different problem perspectives, the required input data can take many forms. Relational-database tools can speed and simplify the process of organizing (and possibly transforming) the input data on hand to match the requirements of a particular analytic need.

We shall not attempt to survey the current state of the art in such environments, except to say that we have found several applications that provide at least some of the capabilities needed for efficient MRMPM work. In particular, we have used the Analytica modeling system for much of our work in this area. We have also experimented with Crystal Ball, @Risk, iThink, EXTEND, and DataView. Each has its strengths and virtues.

3. CASE HISTORY: OUR EXPERIENCE WITH A FAMILY OF MODELS

3.1 The Analytic Context

Much of our experience with MRMPM derives from our analytic work on the military challenge of early interdiction of an armored-force invasion (often called the “early-halt problem”). In this, an aggressor nation (Red) invades one of his neighbors, trying to reach an objective that is a specified distance from his border. Meanwhile, his armored fighting vehicles (AFVs) are being interdicted by Blue, who uses long-range fires in the form of Air Force, Navy, and Army air power, and of Navy and Army long-range missiles.

To analyze this problem, we have constructed and used various models at many different levels of resolution and from multiple perspectives, ranging from entity-level high-resolution simulations to closed-form analytic solutions. Between these extremes, we have constructed several halt models using the principles of MRMPM, and have found them to be useful, both in terms of our analysis and as illustrations of the feasibility and utility of MRMPM.

3.2 Begin at the beginning: Spreadsheets and Closed-form Analytic Solutions

We and our colleagues had done prior work on the halt problem using a theater-level campaign model (JICM) and entity-level simulation using Janus and a local federation. However, we concluded that these were inappropriate for exploratory analysis of the halt problem across a broad swath of scenario space. The entity-level models were too complex and cumbersome, and even the campaign model (which is quite flexible and well suited to certain types of exploratory analysis) is complex and covers too much territory, obfuscating the principal issues of something as narrow as interdiction. Furthermore, we wanted to do our analysis with personal computers. Thus, we began anew, building a simple spreadsheet model focused exclusively on the halt problem. To sharpen our understanding even more, and to make a point about the key drivers being simple, we also derived an even simpler version in terms of a closed-form equation.[4] This expressed the Halt Time (or maximum penetration of the invader) as a function of the various inputs. This said, most of our work at this stage used a spreadsheet simulation, which provided greater flexibility.

From the outset, we had in mind the desire to have MRM/MRMPM models that could also be related to entity-level simulation as a source of “data.” Figure 1 shows the initial hierarchical structure of the processes involved.[3]

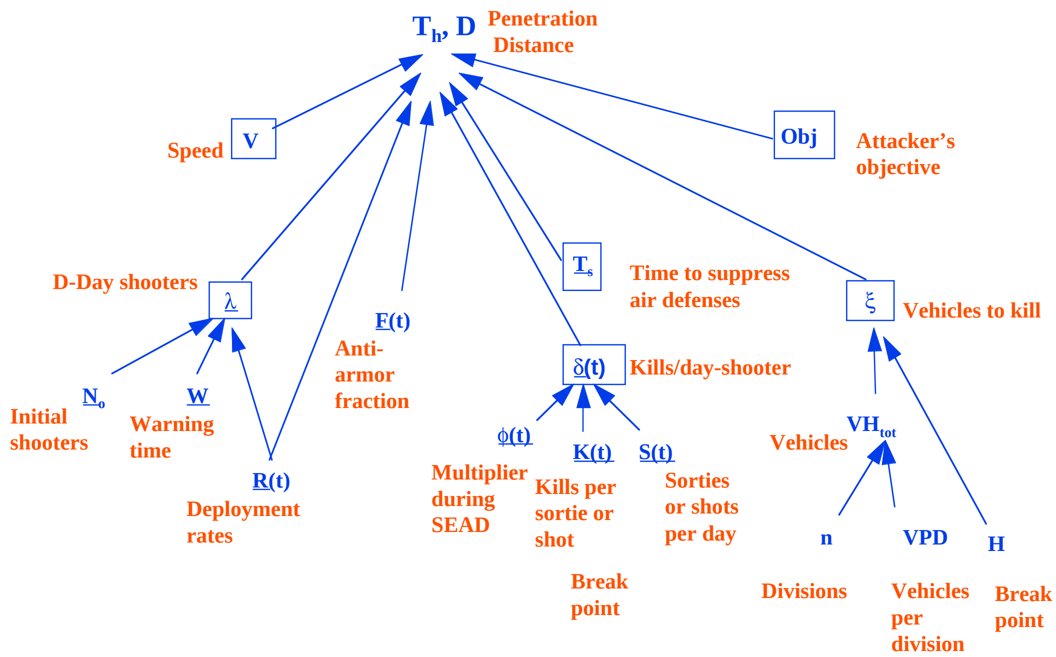

Figure 1: Top-down process hierarchy of spreadsheet halt model.

Readers will note that Blue shooters in the theater on D-Day (the variable λ on the bottom left) can be specified directly or calculated from more detailed variables. Similarly, other variables are in MRM trees. One can also see how multiple perspectives are possible for even a relatively simple model such as this. As an example, a particular user might have a problem with representing Suppression of Enemy Air Defenses (SEAD) effects only by way of a multiplier to effectiveness (φ(t) in the bottom middle). He might instead reason that being in the SEAD phase effectively reduces the number of shooters available to Blue (i.e., he sees fewer shooters flying during SEAD as a shooter-allocation issue, rather than a shooter-effectiveness issue). Although the two representations of the phenomenon can be implemented so that their consequences to outcome are equivalent, they are clearly different perspectives of interdiction processes.

Soon in this work, we began to see a need to move into a programming environment more suited to MRMP modeling. Changing between resolutions and perspectives involved much programming work; even using Visual Basic® programming to facilitate things like changes of resolution, the use of vectors, and so on. The spreadsheet environment was not a natural facilitator of MRMPM implementation. We were beginning to see that our models were maintaining their MRMPM capabilities in spite of our tools, and not because of them. We examined a number of modeling environments, but turned largely in this work to Analytica,® which is marketed by Lumina (www.lumina.com). We used Analytica 1.1.1 for the Macintosh computer.

3.3 Our Analytica Models: EXHALT and PEM

Upon beginning to translate the relatively simple spreadsheet halt model into its Analytica counterpart, we immediately saw benefits, including the environment's visual modeling (an initial attraction, since it facilitated a style of design we had previously followed "by hand") and the ability to handle arrays easily and in mathematically sensible ways. This approach helped us collaborate within our group, and allowed each of us to take a direct hand in the evolution of the halt model, which came to be known as EXHALT (an acronym for <u>ex</u>ploring the <u>halt</u> problem). [4]

3.3.1 EXHALT

EXHALT's starting point was the halt model as it had evolved in spreadsheet form; thus, EXHALT used the hierarchical construct developed for the spreadsheet halt model. One of the first MRM changes made to EXHALT was the decision to represent Blue shooters—just before computing daily vehicle kills toward the top of the tree—as a single "standard shooter" type instead of as a vector of all of the different Blue shooter types. Thus, the scalar concept of the "equivalent shooter" was introduced, which required a substantive change in the EXHALT build. In the vector-shooter version of the model, kills to Red AFVs were obtained by the scalar product of the Blue shooters vector and the Shooter effectiveness vector. If Red AFV kills were to be calculated by the product of Blue equivalent shooters and the effectiveness of an equivalent shooter (both scalars), the model structure would have to be slightly altered at those points. This proved simple to do in the visual modeling environment. The modification did not reduce generality, but *allowed* us to input equivalent shooters directly if we chose to do so (an element of MRMPM). On the other hand, it introduced a concept that was comfortable to some people, particularly analysts, and not comfortable to others. In MRMPM language, we had made a choice of perspective that would improve clarity and credibility for some and reduce it for others.

Once the processes that were in the halt model spreadsheet were incorporated into EXHALT, we began to work collaboratively and visually to enhance the model, adding complicating factors to explore postulated phenomena. Once such phenomenon was the adoption of dash-and-hide tactics by Red. The dash-and-hide concept is that, although the Red army has an assumed movement rate of 40-70 km per day, individual Red AFVs are capable of moving an order of magnitude faster than that for spurts of time, thereby suggesting that it might be possible for Red to cover a day's advance while only being vulnerable to Blue sorties during a small fraction of that time.[4] To account for this effect, we built in a notional multiplier to Blue effectiveness based on Red's postulated ability to adopt these tactics, and on Blue's assumed ability to respond to them (presumably by having good command and control, and surveillance and reconnaissance, and responsive and adaptive attack patterns, possibly including loitering air patrols (CAP)). The visual nature of Analytica allowed this and similar features of EXHALT to evolve as we circulated iterations among our group members by e-mail, each of us making changes.

Over time, we also added other features to EXHALT, such as:

- **Force employment strategies**. This feature allows Blue to attack with either a leading-edge or in-depth strategy.[5] The leading-edge strategy allows "roll-backs" in that if Blue is able to completely shatter the forward x km of a column on a given day, the column's effective advance will be reduced by x km/day. If Blue has enough interdiction capability, x may be greater than the nominal column speed V, in which case the advance will be rolled back by (x-V) km/day. Further, since leading-edge attacks are concentrated on forward units, Blue can stop an entire unit (i.e., remove that unit from the advance) by destroying some fraction of that unit, specified as the "Unit Break Point." We assume that Blue suffers an effectiveness degrade in these leading-edge attacks, since concentrating attacks at Red's front is more difficult than allowing attacks throughout Red's advance. Blue's in-depth strategy suffers no such degrade, but likewise has none of the other advantages of the leading-edge strategy. Which strategy is superior depends on scenario details.
- **Losses to air defenses**. In earlier halt models, we assumed that the losses due to Red's air defenses would never be significant enough to change, to first order, the results of the halt calculations. However, Blue vulnerability (and his valuation of that vulnerability) to Red's air defenses is implicitly included in these models through exogenous parameters such as the "wait time," during which Blue flies at a reduced rate out of concern for Red's air defenses, which are still robust early in the campaign.

Because of this, and to add some measure of the "cost" of certain types of campaigns to Blue, we added a module to calculate Blue's losses to Red's air defenses.

- **Simple agents**. The recognition that many details of Blue force employment depend on the individual scenario faced is an important aspect of this kind of work that is seldom captured in models at any level. Blue's urgency in stopping Red quickly, the importance of intermediate milestones in Red's advance, and Blue's sensitivity to shooter losses all bear on how Blue uses the forces at his disposal in attacks on Red armor. To account for these effects, and to make the point that agents could be included in even simple models, we constructed two agents that adjust Blue's tactics to the situation at hand. These agents are:

  Missile launch agent. Many of Blue's sophisticated precision guided missiles are extremely expensive, and not necessarily plentiful. In recognition that the Blue commander may be unwilling to launch his missiles if he suspects they will be effectively wasted because he did not wait long enough for his C4ISR assets to come on line fully, this agent will withhold the launching of Blue's missiles until some threshold early C4ISR effectiveness is reached, at which point Blue will begin to attack with its missiles.

  Wait-time decision agent. A more intricate example of a decision agent is the optional wait-time decision module. In this agent, the Blue commander estimates various campaign outcomes (particularly, estimated Red penetration at halt, and total Blue losses to Red's air defenses) as functions of how long he flies at a reduced rate due to relatively robust Red air defenses at the beginning of the campaign. Based on those estimates, he chooses a wait time that best matches his criteria for success, using iso-utility tradeoffs between Red penetration distances and losses of Blue's air forces, which are input in the form of a decision table. While this agent started out conceptually simple, its programming implementation was difficult and complicated in practice, as it involved aggregating relatively detailed EXHALT input data and generating "look-ahead" estimates for losses and penetration (we used some of the closed-form analytic solutions for that purpose).

As the features described above (and others) were added to EXHALT, the model's structure evolved from that shown in Figure 1 to the more complex structure in Figure 2, which is a hand-drawn diagram. This diagram can be thought of as the data-flow structure of EXHALT. Note how interactions have emerged between diagram branches; some of these interactions are due to representation choice (the calculation of equivalent shooters requires information about the relative effectiveness of each shooter type, which comes from the effectiveness branch), and others are real, phenomenological interactions (the presence of highly capable, stealthy shooters early in the halt campaign may change Blue's strategy about when to begin attacking, and the early effectiveness of C4ISR assets affects the decision of when Blue begins to use his missiles).

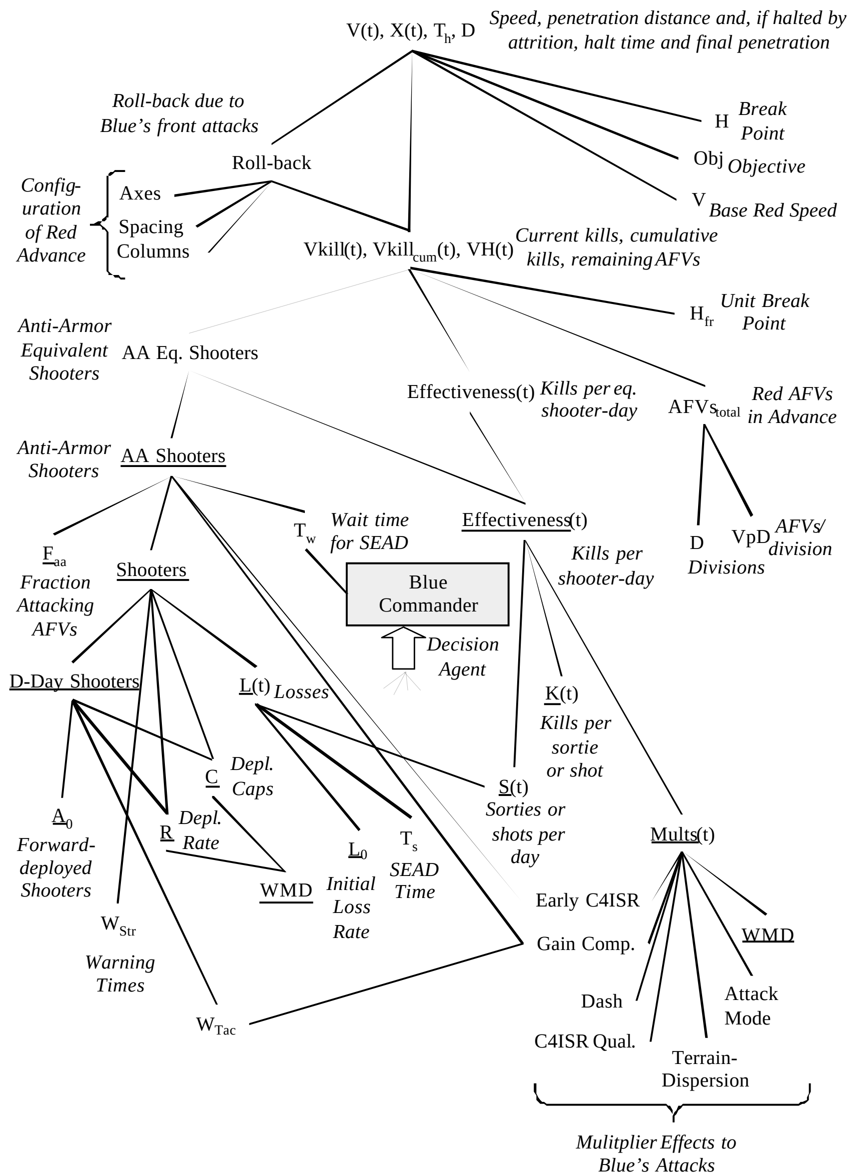

Figure 2. Data-flow diagram of EXHALT.

We constructed Figure 2 for documentation because no such overview could be seen from computer-generated graphics. Analytica's visual programming is (and should be) hierarchically modular. However, this means that one cannot always see the whole. Figure 3 is an example of what one sees. In diagrams such as this, rectangles denote decision variables (inputs), ovals represent stochastic variables, rounded rectangles are intermediate variables, and rounded rectangles with bold outlines are modules. Interactions with variables outside the module denoted by this diagram are indicated by tiny arrows pointing into the left of or out from the right of variable nodes. So, in Figure 3, the node WMD Effects, among others, has effects both within this diagram (represented by the arrow pointing to another node in the diagram) and on variables not shown in the diagram (indicated by the tiny arrow on the right of the WMD Effects node). In this case, for example, WMD Effects also degrades Blue's effectiveness (sortie rates are decreased). Such visual, guaranteed-correct displays are extremely useful for both development and communication. As noted, however, one sometimes wants to see the whole (as in Figure 2).

These interactions do not mean that EXHALT has ceased to be an MRMPM-designed model. Rather, it means that simple truncation of branches, as was possible in the Figure 1 model, now introduces some error into the aggregation (or else the scripting or otherwise parameterization of these more aggregate-level variables must implicitly take these interactions, at least in an approximate way, into account). Alternately, a "hybrid" branch-trimming exercise may allow for some aggregation without losing what a user may regard as important interaction effects. As an example, the missile launch decision is a function of Blue's "gain competence" and "early C4ISR" effectiveness multipliers. A user might wish to consolidate the remaining effectiveness multipliers into an aggregate multiplier, while keeping the effectiveness degrades used in the missile launch decision explicit. This would maintain the model's representation of the interaction, while decreasing the dimensionality of the user's input set.

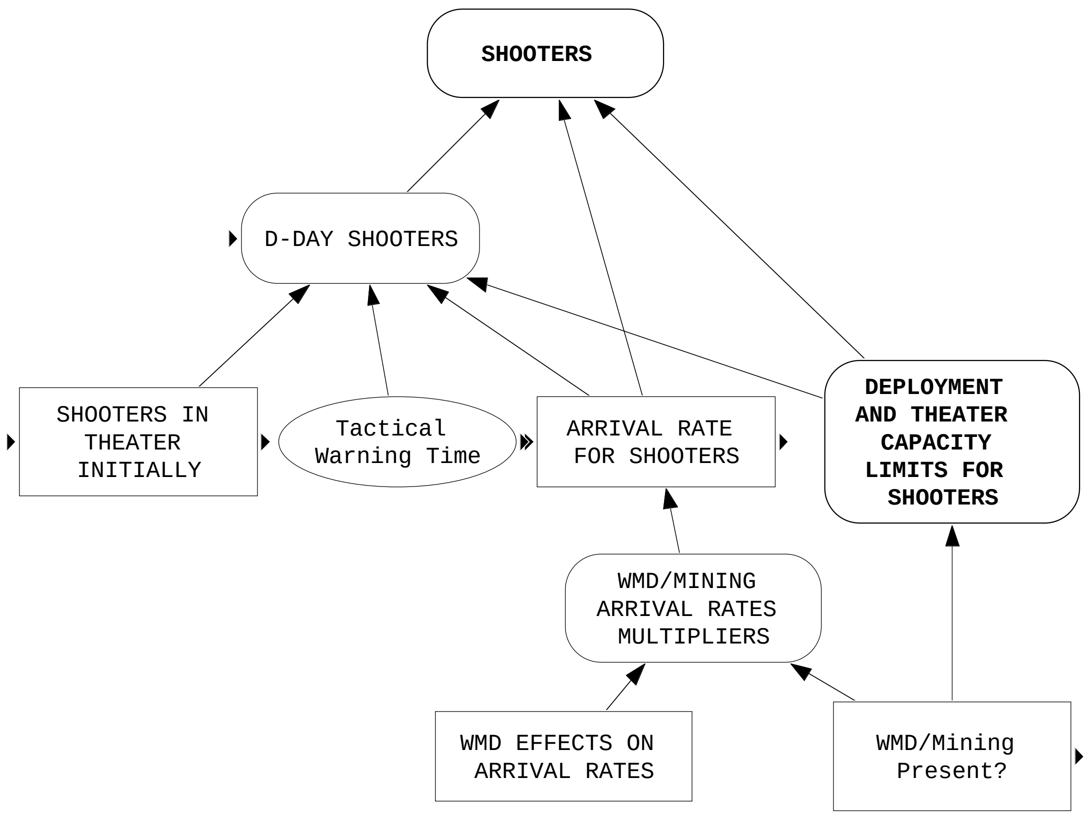

Figure 3. Analytica diagram of data flow for Blue Shooters calculation.

As a matter of practice, to exploit and illustrate EXHALT's MRM structure we decided to implement three user modes of variable resolution into the model itself. One mode would be the full, highest-resolution version of EXHALT, with inputs at the lowest levels of the data-flow branches in Figure 2. The second mode, for illustrative purposes, would trim the forward-deployed-shooters and warning time nodes from the D-Day shooters calculation, and have the user input directly the number of shooters of each type present on D-Day (notice that this does not reduce the number of input variables needed, as warning time is used in the Blue effectiveness calculation, and deployment rate is used further up in the Blue shooters branch). In addition, this second mode turned off the Wait-time Decision Agent and had the user specify the wait time as an input parameter. The third resolution mode was intended to be a very truncated version of EXHALT, in which the Blue shooters and Blue effectiveness branches were severely snipped, and branches parameterizing Red's advance were consolidated (e.g., Red AFVs was input instead of Red division and AFVs per division). What was left is shown as the simpler diagram in Figure 4.

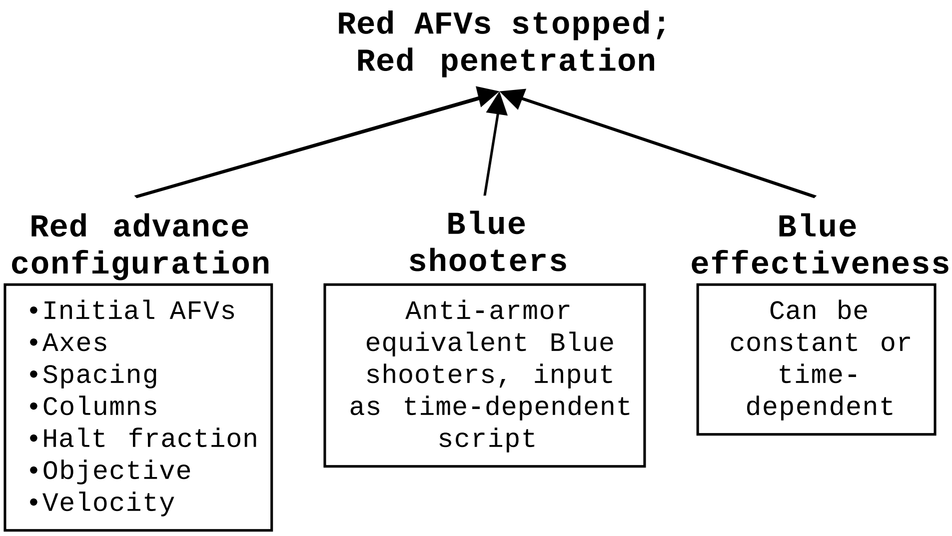

Figure 4. Data-flow diagram of the "truncated" user mode of EXHALT.

Although we knew from the outset that implementing alternative user modes added complexity and could not be pushed very far, we were disappointed to learn that the complexity problem started *very* soon. Because of the interactions that exist in the full model, performing truncation in such a way as to make the user modes universally useful and to render the approximations made for various modes allegedly transparent, the encoding of the various modes quickly became rather complicated. Further, even when we tried to take advantage of these user modes within our own analytic work, we found that none of the representations did precisely what we needed. Initially, we attributed this to a failure on our part to parse the problem correctly. Upon further contemplation, though, we came to the conclusion that, because of varying perspective needs of different users (and even the same users with different problems), the "design it right the first time" ideal is impractical.

Modelers and analysts (including we) have said this for a long time, but we had made an "old college try" to prove otherwise, and failed. Rather, we have reaffirmed the idea of using modularized "building blocks" to build MRMPM structures that can be easily *adapted* and *evolved* to meet the needs of a particular analysis. This means that the model used in a particular application must likely be "assembled" at the time, rather than merely taken off the shelf. The building blocks, hopefully, *would* be on the shelf. Some of this philosophy, of course, is eminently familiar to anyone who has done complex system work or worked with sophisticated simulation environments, but we had hoped that for relatively simple models more stability and flexibility could be built in. The reason for this hope is that clients—especially in government or other large organizations—very seldom want to be provided "environments" and "building blocks." They typically want "commodity models." They may want transparency, but they don't want to make changes. Our conclusion, however, is that the need for building-block-and-assembly methods often becomes important rather quickly—i.e., even with relatively simple models.

Fortunately, visual, interactive modeling environments make this adaptation relatively painless, especially with good modular designs. Again, the alternative is to build every significant perspective into one's models from the beginning, which is practically impossible from a coding viewpoint, even if one assumes that all possible future needs could be articulated. This building-blocks approach yields a much more efficient process, that of extending the model family (while maintaining mutual consistency within the family) to address the needs of a particular analysis. The price, however, is that it requires changes of model and code, not merely changes of data. Even good modelers can make mistakes when making such changes and large organizations know well how serious errors can be made in "simple" spreadsheets for accounting.

There are a few things we have done, though, to make EXHALT useful for more than one purpose. For one, we have implemented switches to allow users to turn certain features (losses, the wait-time decision agent, the calculation of the aggregate effectiveness multiplier from higher-resolution effects) on or off. This was not as easy as expected. Due to interactions, the "turn off losses" switch also turns off the wait time decision agent, since without a loss calculation, the tradeoff analysis performed by the agent between losses and Red penetration is meaningless. Also, the truncation of the aggregate effectiveness multiplier branch had to be done in the "hybrid" method described above. Even when MRMPM ideas are simple conceptually, they are often difficult to implement due to constraints in the programming environment.

3.3.2 A word about the Precision Guided Munition Engagement Model (PEM).

Having developed a high-level halt model (EXHALT), we sought to understand relationships between it and entity-level simulation done by some of our colleagues. The results of that simulation were surprising and not well understood. This caused us to develop PEM,[6,7] which provided an explanation of results that could be extended to many other situations than the ones treated at high resolution, while being calibrated to the high-resolution work.

PEM is itself a relatively simple Analytica model, but it includes a level of detail in representing precision guided munitions that is inappropriate for direct integration into EXHALT. Thus, we studied the phenomenology of PEM and built a "Repro model." Consistent with the philosophy of our prior work, however,[3] we did so using phenomenology and some formal mathematics to motivate the form of the model. That is, we used regression methods, but only after deciding on a phenomenologically sensible, and very nonlinear, form of the model. The result (called Reproduction of the PGM Engagement Model, or RPEM) does not have the stochastic richness of PEM, but it is otherwise quite useful and conceptually straightforward. RPEM is implemented in EXHALT as a "helper" application within the terrain-dispersion effects module to assist users in setting reasonable values for the terrain effects multiplier for each shooter type.

4. LESSONS LEARNED

Having spent considerable time incorporating MRMPM principles into our modeling work, both to demonstrate that it can be done and because of its usefulness and necessity in the kind of exploratory analysis we have done, we have found it interesting to step back and ponder what we have learned about the environments available (and desired) for MRMPM development, and what we have learned about the models themselves.

4.1 Lessons Learned Regarding MRMPM Development Environments

Attributes. The list of attributes described above (visual modeling, interactive languages, etc.) is a good checklist of capabilities to look for. Throughout our experience, when the attributes for MRMPM modeling listed above were present, they greatly facilitated our MRMPM development efforts; when they were absent, they were sorely missed. Our experience with Analytica has been good, although it has weaknesses for simulation and lacks some capabilities we would like. The other tools mentioned earlier all have their own advantages and shortcomings. Indeed, they were all good tools.

Customization is king. The alternative user modes with which we experimented worked to a point, but that point came much sooner than we expected. At-the-time tailoring is essential for building useful models for different resolutions and perspectives. This philosophy has many implications for tool development:

- Foolproofing tools to ensure users understand the effects of their changes as they propagate through their models.
- Configuration control tools to help users track changes to their models as they evolve, and to help ensure that as models evolve, they remain consistent with their brethren (i.e., other models in the family of models). These might include conventions to help with the problems of variable naming and level placement, alias methods to permit use of short-form names at all times, even while keeping track of their full names and assuring full-name uniqueness within the overall model family, or tools to permit changes to be tentative until approved by the collaborative group or model owner.
- Improved graphical user interfaces (GUIs) to allow users to splice and attach building blocks at different levels with even less trouble than in Analytica.
- Mechanisms for going backwards in model development, like the “Undo” or “Restore” command in many office automation and graphics applications, to allow developers to try new approaches or features without the fear of losing handy access to earlier versions

The value of “high-level” languages. As has been proven so often in the past in this and other areas, high-level languages aren’t so high-level and easy when the problem becomes moderately complex. Though data flows and processes are much easier to trace in Analytica than in spreadsheet-type models, EXHALT is *not* always easy to understand (as we discovered during some of the rougher phases of its development, much to our chagrin). The diagrams in EXHALT are incomplete once the model has been modularized, in that one cannot see all the effects of the variables displayed (although, as mentioned, tiny arrows warn users that interconnections exist). Even the code itself, instead of being wonderfully simple, is obfuscated with ugly syntax dealing with array operations, which are clean and simple only when all the terms in an algorithm deal with entire arrays. Otherwise, one sees user-hostile syntax with words such as “slice” and “subscript”, and multiply-nested “if…then” statements. A real need for development is to find ways to improve these situations. A few suggestions include:

- Multiple-display screens to help ensure that interconnections are always accounted for.
- Improved and more flexible syntax to allow the cleaning up of complex and unintuitive algorithms.
- Graphics and visuals to explain what obscure array operations mean to help users and builders alike in creating and using MRMPM constructs.

Tools for building agents. Building even small agents within models can be very difficult in current modeling environments. Tools such as decision tables, utility maximizing functions, and the ability to run simulations-within-simulations would help in this regard. Some agent-building tools are available in the community (e.g., SWARM), some remarkably sophisticated optimization tools exist in spreadsheets (notably Microsoft Excel’s “Solver”), and several companies have more complex optimization tools that employ genetic algorithms, but we are not aware of anything integrated and appropriate for our purposes here.

Search methods. Much better tools are possible and needed for exploratory analysis.[1,2]

4.2 Other Conclusions

With regard to what we have learned about the process of MRM/MRMPM building and design itself, we can close with a few points:

Use building blocks. One of the key assertions that we make is that these models need to be created with building-block structures and assemble-at-the-time methods—especially when dealing with systems of systems. This facilitates future customization to individual users and studies. Simple and easy-to-use languages and development tools are necessary to make the process accessible to a reasonable cross-section of the analytic community.

Be willing to experiment. Particular systems and models will seldom be correct at the outset. Even if a tried-and-true, battle-tested model from a past study is re-used, it will most likely need to be "tweaked" to serve the needs of the current analysis. Users should be willing and able to use development environments to evolve and customize their models to their needs, understanding the constraints they must work with to ensure their products still belong in their MRM family.

Learn as you go. The experience of constructing and evolving EXHALT helped us to think about results from entity-level simulation in a systems-level way. This insight led us to try to understand puzzling results from entity-level simulation by constructing PEM. Finally, the construction of PEM gave us the insight into the phenomenology of the terrain/dispersion problem needed to build RPEM so that the PEM information could be incorporated into EXHALT. At each step of the way, having worked on the problem at several levels of detail helped us as we moved on to others.

Remember that you're not alone. Many of the problems faced in the MRM and MRMPM communities are common, for example, in the high-resolution, parallel-processed world of entity-level simulation. Good simulations confront us with many of the same problems as those of operational planners and users of more aggregate-level model families. Data sources are mismatched; questionably-valid assumptions lurk behind input datasets; entity-level output must be aggregated into results that can be analyzed in a sensible way, and needs are different at the time than during earlier planning. By working to address some of these issues at the more aggregate level, we can perhaps gain insights to solve problems at other levels. Conversely, tools and techniques emerging from high-performance computing environments may well be transferable to modeling at the single-processor level, as well.

Overall, though some capabilities are still lacking, we have seen immense development in the tools and techniques available today for designing and implementing MRM and MRMPM. Recognition of the importance of these types of models is likely to result in further progress. While the half-empty/half-full glass may be unsatisfying analytically, we remember the thimble of only few years ago, and we continue to push for the full glass that future environments may bring.

ACKNOWLEDGMENTS

This paper draws on work accomplished for the Office of the Secretary of Defense in several projects over the period 1997-1999. Reports on those projects are included in the references below.

REFERENCES

1. Paul K. Davis and Richard Hillestad, *Exploratory Analysis for Strategy Problems With Massive Uncertainty,* RAND, Santa Monica, CA, draft book, 2000.

2. Paul K. Davis, "Multiresolution, multiperspective modeling (MRMPM) as an enabler of exploratory analysis," Proceedings, *Enabling Technology for Simulation Science IV, Aerosense 2000*, Orlando, Florida, April, 2000.
3. Paul K. Davis and James A. Bigelow, *Experiments in Multiresolution Modeling*, RAND, Santa Monica, CA, 1998.
4. Paul K. Davis and Manuel Carrillo, *Exploratory Analysis of the Halt Problem: A Briefing on Methods and Initial Insights*, RAND, DB-232, Santa Monica, CA, 1997.
5. David Ochmanek, Paul Davis, Edward Harshberger, David Thaler, and Glenn Kent, *To Find and Not To Yield: How Advances in Information and Firepower Can Transform Theater Warfare*, RAND, Santa Monica, CA, 1998.
6. Jimmie McEver, Paul K. Davis, and James A. Bigelow, *EXHALT: an Interdiction Model for Exploring Halt Capabilities in a Large Scenario Space*, RAND, Santa Monica, CA, 2000.
5. Paul K. Davis, James A. Bigelow, and Jimmie McEver, *Effects of Terrain, Maneuver, Tactics, and C4ISR on the Effectiveness of Long Range Precision Fires: a Stochastic Multiresolution Model (PEM) Calibrated to High-Resolution Simulation*, RAND, Santa Monica, CA, 2000.
6. James A. Bigelow, Paul K. Davis, and Jimmie McEver, "Case history of using entity-level simulation as imperfect experimental data for informing and calibrating simpler analytical models for interdiction," Proceedings, *Enabling Technology for Simulation Science IV, Aerosense 2000*, Orlando, Florida, April, 2000.