

AD-A260 943



2

A RAND NOTE

1992

DTIC
ELECTE
FEB 22 1993
S C D

Variable-Resolution Combat Modeling:
Motivations, Issues, and Principles

Paul K. Davis, Reiner K. Huber

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

93-02969



copy

RAND

98 2 16 099

The research described in this report was sponsored by the Defense Advanced Research Projects Agency under RAND's National Defense Research Institute, a federally funded research and development center supported by the Office of the Secretary of Defense and the Joint Staff, Contract No. MDA903-90-C-0004.

RAND is a nonprofit institution that seeks to improve public policy through research and analysis. Publications of RAND do not necessarily reflect the opinions or policies of the sponsors of RAND research.

A RAND NOTE

N-3400-DARPA

Variable-Resolution Combat Modeling: Motivations, Issues, and Principles

Paul K. Davis, Reiner K. Huber

Prepared for the
Defense Advanced Research Projects Agency

DTIC QUALITY INSPECTED ?

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

RAND

Approved for public release; distribution unlimited

PREFACE

This Note is a "think piece" on variable-resolution modeling developed as part of a project on military science sponsored by the Defense Advanced Research Projects Agency. The work has been accomplished in the Applied Science and Technology program of RAND's National Defense Research Institute, a federally funded research and development center sponsored by the Office of the Secretary of Defense and the Joint Staff. Comments are welcome and can be sent to Dr. Davis in RAND's Santa Monica office or, by electronic mail, to Paul_Davis@rand.org. Dr. Huber is a professor at the Armed Forces University in Munich, Germany.

SUMMARY

The ability to do a cross walk among levels of resolution is important to a wide range of military and defense-planning activities such as systems analysis and policy analysis, war gaming, operational planning, and the development of a coherent military systems science. There are strong logical linkages among these activities, linkages that have been inadequately appreciated in the past. For example, systems analysts serving policymakers must suppress most details and deal with aggregated descriptions, but to understand which details to suppress they need to appreciate real-world military operations and some relatively complex phenomena. Commanders working amidst massive uncertainty to choose among alternative concepts of operation must also take a relatively aggregated view of events, but must thoroughly understand detailed operations, be able to conduct spot checks, and in some instances specify specific low-level missions that bear directly on the success of higher-level operations. And, as a final example, those concerned with high-resolution studies involving weapon-on-weapon interactions need sometimes to see their work in higher-level contexts so as to judge the relative significance of various capabilities and vulnerabilities. Ultimately, workers, whether they be military scientists or commanders, need to appreciate both forests and trees. This requires being able to change perspectives as appropriate and, in particular, being able to view problems at different levels of resolution.

Although one might at first imagine that high-resolution modeling would solve all problems (one could aggregate outputs as necessary), the reality is that even with unconstrained computer power we would need aggregated (low-resolution) models. High-level problem solving and decisionmaking *require* simple, easily used, understandable, and explainable models to frame and discuss issues and to use for exploratory analysis and aggregated uncertainty analysis. This is especially the case when uncertainties are massive, which is common in higher-level decisionmaking. Further, real-world empirical and judgmental data come at different levels of resolution. We need models to accommodate historical data on divisional movement rates and doctrinal concepts of operation as much as we need models to accommodate test-range data on single-shot kill probabilities. That is, we need to develop our models to use knowledge from the top, middle, and bottom levels of detail. We believe, then, that variable-resolution modeling—i.e., modeling that anticipates and designs for changing resolution in applications—is *essential* in improving the state of the art in military analysis and other activities using combat models. Variable-resolution issues

are also at the heart of many substantive problems affecting model interoperability and re-usability (see also Bankes, Davis, Hillestad, and Shapiro, forthcoming).

Currently, it seems that what is described as variable-resolution modeling takes one of three forms:¹ (a) having *selective viewing*, in which the underlying model operates at one resolution, but observers are provided one or more lower-resolution views through the use of aggregated displays; (b) having *alternative submodels* with different resolutions and switches determining which models are used; and (c) having *integrated hierarchical variable-resolution* modeling (IHVR) in which higher- and lower-resolution versions of important processes (e.g., attrition processes) are designed to be in a clear, natural, and well-defined hierarchical relationship, in which case one can change, in steps, the level of resolution at which the model operates by deciding how far down the hierarchy of processes to go before specifying inputs as data rather than as outputs of subordinate processes.

Selective viewing and alternative submodels are much more common than IHVR. They are very useful, and designs employing these concepts should be encouraged. At the same time, they often involve problems such as implicit dependence on model details, a variety of inconsistencies, complexity, and confusing changes of representation as one changes resolution. The problems become acute when the models are used by someone other than their developing organizations (e.g., in distributed war gaming or attempts to "re-use" software).

We believe that the full power of variable-resolution work is often best achieved through IHVR, for which one can have either a hierarchical single model or an *integrated* hierarchical family of models (the underlying design problems are much the same). It is important to distinguish between IHVR and most current "hierarchical families" of models, which are not truly integrated but rather are the result of an attempt to use together models that were designed independently by teams with different objectives and perspectives. Such current families are often very complex, have significant inconsistencies, and provide only a modest flexibility in the degree to which variable-resolution issues can be addressed. Much of this thinkpiece is about IHVR, which seems not to have been emphasized in previous work. We define and describe it with examples. We then discuss what seem to be generic issues and challenges in using it. Finally, we suggest research paths for advancing our understanding of IHVR and facilitating its use.

Our principal conclusions are these:

¹We do not consider here instances in which resolution is changed by merely substituting one set of data for another.

- Much greater emphasis should be placed on the design of variable-resolution models generally (or integrated families of models).
- The best design approach is often IHVR, which permits: (a) predetermined functional relationships for calibration across levels, (b) avoiding inconsistencies when making modifications, (c) consistency of prediction at key points, (d) cognitive smoothness (consistency of representation), and (e) multiple options for levels of resolution. All in all, then, IHVR permits (but by no means guarantees) what may be described as relatively *seamless* variable-resolution modeling.
- Except in simple problems, IHVR designs will require approximations that break or weaken "cross talk" between hierarchical branches. Finding, specifying, and estimating the quality of approximations creating hierarchies should be a principal task of high-level design. There are instances in which breaking or weakening the cross talk is not feasible because of strongly correlated processes, but IHVR is not an all-or-nothing situation. Substantial benefits can be obtained by having only some aspects of an overall system designed with IHVR.
- One should expect designs to be modified with experience, but laying the basis for variable-resolution issues at the outset is very helpful, because introducing unanticipated variable-resolution features later is often difficult and fraught with dangers. This strongly suggests designing relevant hierarchies early and establishing stubs where detailed models will later be developed. Rapid prototyping and related exploration are still very much possible, but should follow a structural design that may take days or weeks. In our experience, such prior designs, when accomplished by people with considerable domain expertise, hold up well over time.
- Unfortunately, current software environments are not very helpful in developing variable-resolution models. There are many difficulties related to incomplete specification, side effects, ambiguities in control structure, conceptual "modules" being distributed throughout code, and confusing discrepancies between the conceptual model and the implemented program.
- Advanced modeling languages and environments could greatly mitigate these problems. There is particular value to high-level English-like languages that narrow the gaps among analysts, modelers, and programmers while encouraging good naming of objects and processes; data dictionaries; graphical interfaces that move us toward graphically defined specifications in the form of entity structures, data-flow diagrams, and state-transition diagrams; object-oriented modeling methods; and highly interactive environments in which modelers can readily make changes and experiment with them.
- With such advanced environments it will be easier to strike a balance between good initial design and rapid prototyping. That is, it should be easier to develop good hierarchical designs early, while maintaining flexibility for subsequent adaptations.

We also conclude that there is no organized understanding of variable-resolution issues in the community and that it is appropriate to convene a series of conferences to explore the issues in depth. RAND held a preliminary workshop in the fall of 1991 and will

organize a full conference for the late spring of 1992. An objective of that conference will be to bring together people from diverse backgrounds, because variable-resolution issues arise in many disciplines and there is much that can be learned by cross-fertilization. Among the subjects to be treated are the degree to which object-oriented methods can help in variable-resolution design, how workers in different disciplines have dealt with variable resolution, and what kind of generic software tools would be the most useful.

ACKNOWLEDGMENTS

We would like to acknowledge numerous discussions with colleagues Steven Bankes, Richard Hillestad, and Norman Shapiro, with whom we have been working in the current project. They also provided important comments on the first draft, as did Richard Hundley. Bart Bennett provided a detailed and constructive formal review.

CONTENTS

PREFACE	iii
SUMMARY	v
ACKNOWLEDGMENTS	ix
FIGURES AND TABLES	xiii
Section	
1. INTRODUCTION	1
Background	1
Objectives	2
Approach	2
2. MODEL REQUIREMENTS	3
The Usual View: Different Users Need Different Models	3
An Alternative View: Users Need Integrated Variable-Resolution Models	6
3. TYPES OF VARIABLE RESOLUTION	12
Attributes of Variable-Resolution Models	12
General Types	13
Discussion	13
4. DESIGN OF INTEGRATED HIERARCHICAL VARIABLE-RESOLUTION MODELS	22
An Illustrative Problem Showing the Value of IHVR	22
Complications in Designing for Hierarchical Variable Resolution	26
Other Issues in Considering IHVR	37
5. CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH	44
Conclusions	44
Suggestions for Further Research	45
Appendix	
A. SYSTEMS ANALYSIS AND NONMONOTONIC BEHAVIOR	47
B. HISTORICAL BACKGROUND	50
C. AN EXAMPLE OF CONSISTENCY IN AGGREGATION	53
BIBLIOGRAPHY	57

FIGURES

2.1. Interactions Among Systems Analysis, War Gaming, and Military Systems Science	10
3.1. Possible Low-Resolution Display for a Higher-Resolution Simulation	14
3.2. Effect of Hidden Variables in Aggregated Display of Higher-Resolution Simulation	15
4.1. A Model for Calculating Damage Expectancy	22
4.2. A Damage-Expectancy Model with Integrated Hierarchical Variable Resolution	23
4.3. A Dialog Box for Specifying Level of Resolution	25
4.4. An Example of Nonhierarchical Relationships	27
4.5. A Revised Design Viewed for the High-Resolution Case	28
4.6. A Revised Design Viewed for the Low-Resolution Case	29
4.7. Generic Example of Side-Effect Issues in Variable-Resolution Modeling	30
4.8. Example of a Distributed Model	32
4.9. Comparison of Quasi Code for Low-Level and Incompletely Specified Higher-Level (Low-Resolution) Models	34
4.10. Control Flow for an Override Option	35
4.11. Control Flow for a Bypass (Substitution) Option	36
4.12. Alternative Perspectives Reflected in Hierarchical Designs	39
4.13. Example of Political-Model Variable Resolution	40
4.14. Example of Variable Resolution in Command-Control	41
4.15. Variable Resolution in the Generalized Force Ratio Model	42
A.1. Monotonic Behavior of Outcome vs. Capability for Options A and B	49
A.2. Inconsistency in Systems Analysis Due to Nonmonotonic Model Behavior	49
C.1. Altitude vs. Time for Separate Falling Bodies	55
C.2. Aggregated vs. Exact Model Predictions of Average Altitude vs. Time	56

TABLES

2.1. Stereotyped Comparisons	4
2.2. Comparisons in the Broader View	9
3.1. Variants and Attributes of Variable-Resolution Model Types	20
B.1. Types of Zoom Capability	52

1. INTRODUCTION

BACKGROUND

The Department of Defense (DoD) has recently concluded that modeling and simulation will be a *fundamental* part of its activities in the years ahead—in functional areas that include: education, training, and operations planning; research and development; production and logistics; test and evaluation; and analysis. Part of the DoD's vision for using modeling and simulation calls for ambitious networking in which geographically distant and functionally different organizations routinely work together, relying heavily upon a wide variety of models differing in resolution and in many other respects as well. This networking will include distributed war gaming, planning, and a myriad of other functions. As an example, exercise participants will include, simultaneously: air force pilots actually flying their aircraft, soldiers "fighting" tank battles in realistic simulators, general officers making and communicating their decisions within the real-world command and control system of a joint task force, and analysts using models to answer "What if?" questions, either to help inform decisions or as part of an effort to maximize lessons learned from an exercise. Indeed, none of this is even futuristic at this point. More futuristic is the vision of such activities fitting together so perfectly that the differences between simulated reality and reality itself merge to a very significant degree.¹

All of this will require an unprecedented degree of interoperability among models. In the past, models have typically been designed for specific purposes, with little regard to how they might be used with other models. Today, that is changing rapidly. Further, some of the traditional distinctions among classes of activity are beginning to break down as the DoD increasingly emphasizes such concepts as learning and training with the same tools one would use to fight.

One consequence of these and related trends is an awakened interest in the capability to vary levels of resolution readily—either within a single model or within a family of related models. Such capability is at the heart of achieving the DoD's vision for modeling and simulation.

¹This vision has been communicated in a variety of forums by the Director of Defense Research and Engineering (DDR&E) and the Defense Advanced Research Projects Agency (DARPA). Information can be obtained from the Defense Modeling and Simulation Office (DMSO) within the office of the DDR&E. Some of the material in this Note is taken from a DMSO paper entitled "Background Information for Industry Day," February 1992.

OBJECTIVES

It is with this background, then, that the objective of this Note is to introduce and discuss with concrete examples the general subject of variable-resolution modeling, addressing such questions as "What is it?" "Why might we want it?" "What forms does it take?" "Why is it nontrivial?" "What principles might be applied in attempting it?" and, finally, "What additional research would clarify the subject and improve our ability to accomplish variable-resolution modeling in its various forms?" Importantly, this Note should be considered a "think piece." Its purpose is to motivate interest, identify issues, suggest some new ideas, and suggest what should be done next. The Note is part of a continuing effort sponsored by DARPA.²

APPROACH

Our approach will be as follows. Section 2 discusses why differing levels of resolution are needed in combat models and why many model users have not fully appreciated this need. Section 3 describes several forms that variable-resolution modeling can take. Section 4 illustrates the concepts with concrete examples and identifies design issues, particularly the value of hierarchical modeling and the tension between anticipating needs in an initial design and charging ahead with a rapid prototyping approach that maintains flexibility. Finally, Section 5 summarizes conclusions, the most important of which bear on what needs to be studied in more depth and what types of software tools may need to be developed to improve capabilities for variable-resolution modeling and analysis while maintaining a high degree of flexibility for continued model adaptations.

²Portions of the Note draw on material prepared originally in 1988 as part of the development of the RAND Strategy Assessment System (RSAS). The Note also reflects an attempt to follow up on problems identified in Davis and Blumenthal (1991).

2. MODEL REQUIREMENTS

THE USUAL VIEW: DIFFERENT USERS NEED DIFFERENT MODELS

The conventional view (from which we depart below) is that models should be designed to match the needs of *specific* users. As a result, models may differ with respect, for example, to

1. the *questions asked* (e.g., "What if?" vs. "Is it possible for . . . to happen?" vs. "How could I get from here to there?" vs. "What is the least-cost way to . . . ?")
2. *generality* (e.g., one wants theory and doctrine to be based on general concepts, often described at a high level of abstraction, whereas operations plans must be highly specific)
3. the *treatment of uncertainty* (e.g., not at all, by considering some alternative scenarios, or by extensive parametric analysis)
4. the *variables treated* (e.g., those relevant to influencing defense programs vs. operational strategy vs. the education of officers)
5. the *data available* as input (e.g., weapon-on-weapon kill probabilities vs. historical data on average divisional rates of advance in successful large-scale operations)

It follows that we need models differing, among other things, in resolution. Further, different organizations and different classes of activity focus on particular levels of resolution rather than others. Their models reflect this and those working in organizations often come to think exclusively in the terms reflected in their organizations' models.

These differing needs have led to what amount to different modeling paradigms. Table 2.1 shows stereotyped comparisons among three illustrative classes of activity requiring models: *systems analysis* (or what is often called *policy analysis*), *war gaming* for both education and the development of operations plans, and development and exploitation of *military science*. Although many exceptions to the stereotypes can be found (more on this below), drawing the contrasts starkly is a useful place to begin.

Table 2.1
Stereotyped Comparisons

Uses	Systems Analysis	War Gaming	Military Science
Target audience	Policymakers	Future or current commanders	Students, analysts, developers of doctrine, military planners . . .
Time frame	Future	Current and near term	All
Vary strategy?	Often	Seldom	Often
Vary scenario?	Often	Seldom	Often
Vary capability?	Often	Seldom	Often
Focus	Choice of alternatives	Training and plan building	Phenomenology
Breadth	Great	Modest	Great
Search style	Breadth first (compare top-level alternatives, described simply)	Depth first (investigate particular strategy in depth)	Eclectic
Attitude toward models	Patch together as needed	Do they behave? Do they represent entities of interest?	Seen as representation of knowledge and mechanism for exploration
Perceived need to understand models	High	Modest (black box deemed ok)	High: models represent theory
Perceived need for aggregation	High	Depends on rank and style	High: all science must deal with micro and macro worlds
Perceived need for detail	Selective (consider detail as <i>necessary</i>), not for its own sake	High or low depending on war game	High: all science must deal with micro and macro worlds

Systems Analysis

In this depiction, systems analysts/policy analysts¹ work on policy-level questions and are concerned primarily with the future (e.g., in advising on the relative future value of alternative weapons systems in competition for limited dollars in the defense program, or in estimating possible military consequences of one or another total force structure or possible arms control agreement). Ideally, they approach their work with a broad *multiscenario*

¹What we call systems analysis here is a combination of what in Germany is called systems analysis, decision support analysis, and policy analysis. In the United Kingdom, the terms policy analysis, systems analysis, and operations analysis are almost interchangeable. Unfortunately, systems analysis also has a very different meaning, particularly in the software community, where it is associated with the design and development of complex computer programs.

analysis that explores the consequences of varying strategies, political-military scenarios, capabilities, and forces (see, e.g., Davis, 1986; Davis, 1988a; and Bankes, 1992). They emphasize breadth of analysis over depth, and they rely upon low-resolution models and aggregation concepts such as Armored Division Equivalents (ADEs). Systems analysts are vigorously *reductionist*, attempting to eliminate all unnecessary detail so as to illuminate the central issues that should be considered by policymakers.²

War Gaming

Standard war gaming, by contrast, is most often employed for education and training (e.g., at the Joint Warfare Center, the Warrior Preparation Center, and various military commands).³ Here the emphasis is typically on acquainting participants with the elements of what they or the people they serve from staff positions will eventually command, standard operations, and standard problems. There is often great emphasis on verisimilitude and depth, especially in training contexts. Models are often required to represent all the entities of interest to participants and to reflect rather accurately and perhaps precisely how those entities would behave in standardized scenarios. There has often not been much emphasis on considering variations of strategy, forces, and the like—if for no other reason than because time is limited and complexity is overwhelming (especially with large numbers of participants and detailed war gaming).

Military Science

As discussed in Davis and Blumenthal (1991), those pursuing military science, whether or not by that name, have yet another set of needs. They are interested in phenomena and *theories* that help explain phenomena and inform choices about how to formulate doctrine and strategy. In practice, most military scientists focus on phenomena at only one level of detail (e.g., weapons, small units, maneuver units, the operational level, or the strategic level). Thus, they tend to focus on particular models and levels of resolution. As a whole, however, the community of *de facto* military scientists employs models of all types.⁴

²For some purposes, they also seek “well-behaved” and monotonic models, as described in Appendix A, which discusses the potential conflict between that and realism. For classic works on systems analysis/policy analysis, see Fisher (1970), Quade and Boucher (1968), Goeller et. al. (1983), and Hughes (1989).

³For discussion of war gaming at the Warrior Preparation Center, see Rehms (1991) and Allen (forthcoming?). There are, of course, other types of war gaming. Analytic organizations employ war gaming in a variety of ways that include contemplating the potential impact of new technologies or alternative political-military crisis-management strategies.

⁴A partial analogy can be made here to other sciences. High-energy particle physicists seldom are experts in the physics underlying the interaction of molecules or the phenomena of viscous fluid flow. There is, however, an important difference between physical scientists and those working with

AN ALTERNATIVE VIEW: USERS NEED INTEGRATED VARIABLE-RESOLUTION MODELS

Most models have indeed been built to reflect a particular perspective on the world and to help with a particular class of problems to which that perspective is deemed appropriate. As a result, most existing models treat any given phenomenon they represent at a particular level of resolution. It is commonly necessary, however, for users of models to consider different resolutions at one or another point in their work:

- Low-resolution models are needed for (a) initial cuts (innovation and exploration), (b) comprehension, (c) systems analysis and policy analysis (or, more generally, supporting decisions under uncertainty), (d) adaptability, (e) low costs and rapid analysis, and (f) making use of low-resolution knowledge and data.
- High-resolution models are needed for (a) understanding phenomena, (b) representing knowledge, (c) simulating reality, (d) calibrating or informing low-resolution models, and (e) making use of high-resolution knowledge and data.

Importantly, both classes of model—and intermediates⁵—will be needed even as we obtain “infinite computer power.”

The following are more concrete examples of how those using combat models may want to vary the level of resolution in their work: ⁶

1. *Using high resolution to provide a “picture.”* One is conducting an aggregated systems analysis, but wants to review “what’s really going on”—perhaps as a check that the simplified model being used for analysis is not missing key phenomena. Someone using a piston model for a corps’ ground combat might want, for example, to follow some higher-resolution depictions of “representative” battles consistent with the piston model’s data. Afterward, he might conclude that the piston model is adequate as a high-level (low-resolution) depiction, despite there being considerable tactical-level maneuver; or, he might conclude that the piston model is inadequate in the circumstances because of nonlinear combat including flanking operations and deep operations with airmobile infantry and attack-helicopter units.
2. *Using high resolution to establish bounds.* One is conducting an aggregated policy analysis in which reconnaissance is reflected in parameters affecting the reaction time for concentration and counter concentration. One might want to

combat models: the former study their discipline at a variety of resolutions when in school; they study texts that acquaint them, at least to some extent, with a wide variety of perspectives and models. As a result, they understand the value of the perspectives and models, even if they subsequently specialize. Specialists in the different domains of physics can at least communicate with one another. By contrast, those using high- and low-resolution combat models often spend more time quarreling than communicating. There is need to break down the barriers among cultures.

⁵Although many examples in this Note involve only two levels of resolution, the ideas we present apply also to multiple levels.

⁶Appendix B gives some of the historical background on perceived needs for variable-resolution modeling.

employ high-resolution simulations to assess how wide a range should be used for sensitivity analysis.

3. *Using high resolution to calibrate.* One is conducting an intermediate-level simulation of ground combat that depends on killer-victim attrition matrices. Suppose that in the course of the simulation one sees a battle emerging that involves circumstances significantly different from the reference cases against which the matrices were originally calibrated. One might then want to interrupt the basic simulation, conduct a set of high-resolution simulations or war games, recalibrate (probably using both high-resolution results and softer information such as experience and doctrine), and continue.
4. *Using low resolution for decision support.* One is conducting a war game, either with human teams or automated decision models ("agents"). Given the many uncertainties about opponent tactics and the exact state of battle, many decisions must be made on the basis of a low-resolution view. Players or decision models might want to conduct a series of low-resolution simulations to assess the relative goodness of alternative courses of action. These should preferably be consistent with the high-resolution models. In this case, the low-resolution models would be used while the higher-resolution simulation was interrupted, or as specialized subroutines serving decision models (e.g., "optimizing models" that decide on allocation and apportionment of tactical air forces).
5. *Using low-resolution modeling to "generate scenarios."* Suppose one is able to conduct detailed simulations of combat. One may want to use low-resolution multiscenario modeling and analysis to establish test cases to be examined in detail. Or, within a high-resolution war game, one may want to use a low-resolution model to generate and maintain overall political-military "context."

For all of these purposes it is convenient to have the ability, within either a given model or a family of models, to change resolutions. Usually, this is difficult or impossible, because the models were developed separately and simply don't fit together well. The goal here should be "seamless" models, defined as follows:

Seamless variable-resolution models: models that permit changes of resolution while maintaining (a) consistency of prediction and (b) consistency of description or representation.⁷

To illustrate this, suppose one has low- and high-resolution models of ground-combat movements that are "seamlessly related." We would expect the concepts (and variable names) of the two models to be clearly related and the predictions of the low-resolution model to be a good approximation over time of what one would obtain by aggregating predictions of the high-resolution model (at least within a reasonable domain of initial states). For

⁷The subject of consistency is complex. See forthcoming work with colleagues Steven Bankes, Richard Hillestad, Norman Shapiro, and Mario Juncosa.

example, the average FLOT (forward line of own troops) penetration obtained from a low-resolution model should look, over time, much like what one would obtain over time by averaging the penetrations of the multiple FLOTs treated in a high-resolution model.⁸

It is against this background that we should explain the motivation for the work described here. The motivation was essentially two observations:

- *Virtually all classes of users of combat models (analysts, educators, doctrine developers, commanders . . .) need models allowing them to work at different levels of resolution.* This need, however, is often not fully appreciated, nor its implications understood. Indeed, different cultures have arisen in which one or another level of resolution is considered "right" and descriptions at other levels of resolution are used only on an ad hoc basis, almost as a necessary nuisance. It is desirable that the various cultures all come to appreciate the value of alternative descriptions of the same system and their own need to be able to move back and forth among them.
- *There is no recognized set of concepts and design methods to guide the building of combat models to facilitate working at different levels of resolution.* Modelers have used a variety of ad hoc methods to provide some capability for moving across levels of resolution, but it is often and perhaps usually necessary for users to do considerable off-line analysis to assure some measure of consistency and to make sense of results. New concepts and design methods are needed to mitigate this problem.

Table 2.2 summarizes comparisons from this alternative viewpoint, using bold type to indicate where it is different from Table 2.1. In this depiction, a systems analyst can scarcely do a good job unless he understands enough about military operations and military phenomena to guide his reductionism.⁹ Moreover, while systems analysis is often thought of as focused more on choosing among options than on developing strategies, the best options are often mixed strategies and the best options often have many integrated components. Thus, the more sophisticated system analysts find themselves integrating, and find themselves very interested in military operations and military art.

Similarly, as modern theater-level and national-level war games are conducted by serious military commanders, the better commanders will become increasingly interested in the *validity* of the models used to adjudicate consequences of war game decisions. Moreover, they will begin increasingly to construct building-block strategies, to think of operational

⁸As illustrated in Appendix C for an extremely simple physics problem, it is not always intuitively clear whether aggregated equations will be accurate or inaccurate. Furthermore, their accuracy depends on the values of all the relevant variables.

⁹A notorious example of this was the improper use, in the late 1970s, of WEI/WUV scores to "prove" that light infantry divisions were less cost-effective than mechanized forces. Analysts, beguiled by aggregate methodology, trivialized a complex and important issue, and, arguably, got the wrong answer.

Table 2.2
Comparisons in the Broader View

Uses	Systems Analysis	War Gaming	Military Science
Target audience	Policymakers, commanders, developers of doctrine . . .	Future or current commanders; also, analysts and military scientists	Students, analysts, developers of doctrine, military planners . . .
Time frame	Future (also the present when comparing strategies)	Current and near term; and, in policy studies, the future	All
Vary strategy?	Often	Often	Often
Vary scenario?	Often	Often	Often
Vary capability?	Often	Sometimes (especially in support of policy studies)	Often
Focus	Choice of alternatives	Training, plan building and evaluation; also, experimentation with new doctrines, etc.	Phenomenology
Breadth	Great	Moderate (dependent on type of game)	Great
Search style	Breadth first (compare top-level alternatives, described simply)	Depth first (investigate particular strategy in depth) in some; breadth first in others	Eclectic
Attitude toward models	Patch together as needed, but need integrated families	Do they behave? Do they represent entities of interest? Do they capture essence of issues?	Seen as representation of knowledge and mechanism for exploration
Perceived need to understand models	High	Moderate	High: models represent theory
Perceived need for aggregation	High	Depends on rank and style	High: all science must deal with micro and macro worlds
Perceived need for detail	Selective (consider detail as necessary), not for its own sake	High or low depending on war game	High: all science must deal with micro and macro worlds

strategies as options (Hosmer, 1988), and to see their job as making difficult choices under uncertainty guided by estimates of probable and possible effectiveness and cost (in terms of lives lost rather than dollars). Another trend here is to use war gaming in evaluating new doctrinal concepts and potential future systems. For all of these reasons, those using war gaming will increasingly be adopting some techniques of systems analysis.

Figure 2.1 illustrates the connections that should and to some extent do exist among the several activities. For example, policy-oriented systems analysts can produce scenarios studied in depth by war gaming; they can also produce many hypotheses (often based on low-resolution thinking and modeling) to be explored and tested in more depth by war gamers and military scientists drawing upon models and knowledge of many different kinds and resolutions. War games provide, in many cases, the next best thing to empirical data for use by both analysts and military scientists.¹⁰ These quasi-empirical data may take the form, for example, of *likely* tactics by the opponent in response to tactics by one's own side. Military science, of course, should be providing constructs, frameworks, theories, and models themselves.

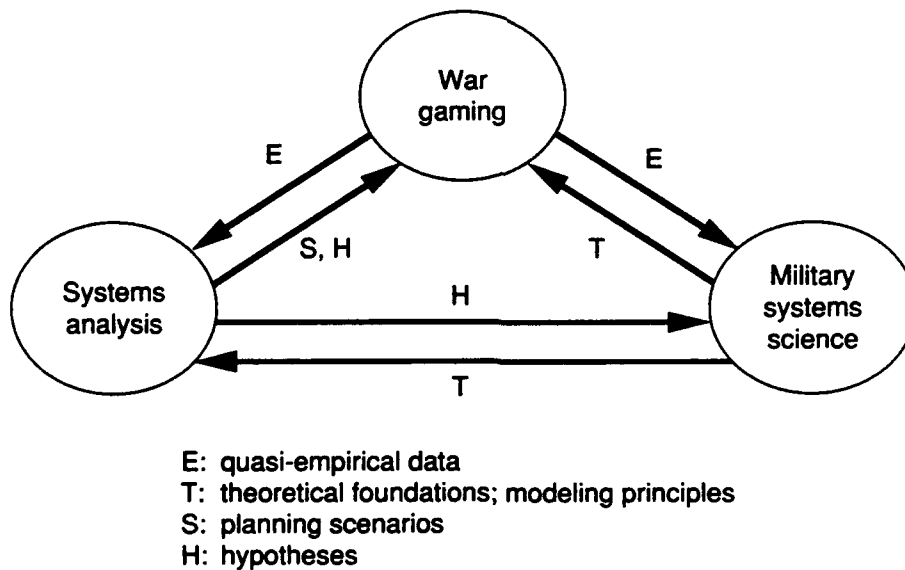


Figure 2.1—Interactions Among Systems Analysis, War Gaming, and Military Systems Science

¹⁰Other data, of course, can be obtained from test ranges, laboratories, and engineering models.

It should be evident from this discussion, then, that there is need for more integrative models representing the best of our military knowledge and flexible enough to serve the needs of the several communities mentioned here. The choice is having different models for different purposes, perhaps with some cross calibration, or having integrated variable-resolution models. We must live with many existing models that were not designed as part of integrated families, but with new models we can do better. Further, we will find it easier to connect existing models if we bear in mind how we wish they had been integrated in the first place. Let us next consider alternative approaches to variable-resolution modeling.

3. TYPES OF VARIABLE RESOLUTION

Variable-resolution modeling: the building of models or integrated families of models in such a way as to permit the user to change the level of resolution at which phenomena are treated or displayed.

ATTRIBUTES OF VARIABLE-RESOLUTION MODELS

By variable-resolution modeling, then, we mean designing models from the start with the notion that users will need to change resolutions. Thus, variable resolution is a goal rather than an afterthought. Further, design should aspire toward "seamless" models as defined in the last section.

Variable-resolution modeling can take a variety of forms, which we describe below. Before doing so, however, it is useful to note that one can characterize a given variable-resolution model in a number of dimensions, particularly the following:

1. *When can resolution be changed (i.e., what are the change points)?* Before running the simulation? At special interrupt points during the simulation? At any point in the simulation? In postprocessing of simulation outputs?
2. *How much flexibility is there?* How many levels of resolution? For how many processes?¹ How long does it take to change resolutions (seconds, minutes, hours . . .)?
3. *Are the changes reversible?* Can one "zoom in" and also "zoom out?"
4. *Are the alternative resolutions contained with a single model, or in a family of models?*
5. *How consistent are the results?*
6. *Given consistency at a calibration point, how well does the consistency extrapolate or interpolate to other points?*
7. *Are the descriptions or representations consistent? Are transitions cognitively smooth?*
8. *If one seeks to fit lower-resolution parameters to outputs of higher-resolution simulation, is it clear how the mapping is to be accomplished?* For a single simulation experiment? For a set of experiments across cases?

Or, combining (5)-(8),

¹By "processes" we refer to model functions and procedures describing phenomena. For example, ground-combat units are subject to attrition processes and maneuver processes.

9. *How well integrated are the depictions at different resolutions? How "seamless" is the variable-resolution capability?*

GENERAL TYPES

Against this background, we have found it useful to characterize models said to have variable-resolution features in three classes or types. After describing them briefly, we will compare the types using the attributes above. The model types are:²

1. **Selected viewing:** *Having the option to work at an apparent low level of resolution by "seeing" only a subset of the variables representing the aggregated concepts of most interest. The underlying simulation, however, is more detailed.*
2. **Selecting alternative submodels:** *Having the option to change resolution by changing submodels altogether—not as described in the third method, below, but by substituting processes. That is, one has low- and high-resolution models of a subsystem (probably developed more or less independently); both exist within a larger model, with a switch allowing one or the other to be chosen in a particular run. Sometimes one invokes a different submodel by "manual intervention."*
3. **Integrated hierarchical variable resolution (IHVR):** *Having the option to change the resolution of the underlying simulation seamlessly, albeit in steps, by exploiting integrated hierarchical design. By choosing to operate at low resolution, one is truncating the simulation at a certain level of detail. By choosing to operate at higher resolution, one is extending the hierarchy of detail.*

Let us next give examples of each of these approaches and describe some generic strengths and weaknesses. Then, Section 4 gives examples. Appendix B provides historical background on how these approaches have been discussed in the past; it also discusses their relationship to what is sometimes called "zooming."

DISCUSSION

Selective Viewing

Selective viewing involves displaying low-resolution information even though the underlying simulation is more detailed. It can be very useful in understanding the essence of what is going on in a complex system and it is often an essential part of verification, validation, and accreditation (VV&A), since many of the most common errors show up quickly when one looks at aggregated results, as do important trends.

²Arguably, a fourth type of variable resolution involves changing the resolution of data rather than the resolution of the model itself. An example here might be changing the richness of a road network along which ground forces are allowed to move and fight. Another example might be shifting from an order of battle with aggregated, notional units to a much more detailed order of battle with specific named units and unit-specific weapon holdings. We do not discuss this class of variable resolution further in this Note.

As a first illustration, suppose one had a simulation model describing ground combat within a given corps, a simulation that measures force strength with "scores" (e.g., equivalent division scores), but distinguishes between FLOT forces and reserves, considers frontage, command-and-control decisions about the reserve fraction as a function of militarily usable frontage, force levels, and so on. To keep discussion simple, consider this model to be "detailed," even though far more detailed simulations exist. This simulation is detailed with respect to an even more aggregated model that would consider only total attacker and defender forces (i.e., not distinguishing among FLOT forces and reserves and not considering frontage).³ In selected viewing for this example, then, one might have a display such as that in Fig. 3.1 showing the principal aggregated variables: A_{tot} , D_{tot} , $F_{tot} = A_{tot}/D_{tot}$, DLR, ALR, and RLR—i.e., the total attacker strength, the total defender strength, the force ratio, the defender loss rate (fraction of force lost per unit time), the attacker loss rate, and the ratio of loss rates.

This is a useful display for many reasons. First, the viewer can see the aggregate situation. In this case the attacker has a substantial 4.2:1 advantage in force ratio and is "winning" as shown by the ratio of loss rates being 0.69, which is less than the breakeven ratio of 1. Second, the viewer can check quickly on the simulation's aggregate reasonableness. It shows the defender taking losses (in equipment-based strength) at about 6 percent per day. The viewer can mentally compare that to a historical figure that he uses as a rule of thumb. A 4 percent attacker loss rate at the corps level is "in the ballpark," as are the other figures. Thus, the selective viewing permits some tests of face validity. In practice, many of the most common programming and modeling errors have big effects on aggregated results, so this type of aggregated checking is very useful.

In the context of an interactive war game with the viewer playing a defending corps commander, the display would be quite enough to cause him to ask for theater reserves or

Time	A_{tot}	D_{tot}	F_{tot}	DLR	ALR	RLR
4	10	2.40	4.2	.06	.043	.69

Fig. 3.1—Possible Low-Resolution Display for a Higher-Resolution Simulation

³Spreadsheet versions of the two models alluded to here were developed by one of us (Davis) for pedagogical purposes in illustrating aggregation and calibration issues. They describe score-based close combat in a single sector, using the Lanchester square law and the 3 to 1 rule for calibration. More elaborate versions include flanks, reinforcements, factors for terrain and type of battle, and aircraft; also, the scalar scores can be replaced by vectors and the kill coefficients replaced by killer-victim matrices. Model details are irrelevant here.

consider a withdrawal—he would not need more detailed information to know he was in trouble (except, perhaps, for information on whether either side has theater reserves en route to his sector).

Despite its usefulness, selective viewing has a number of distinct problems or limitations:

- **Hidden variables.** Figure 3.1 suggests by its structure that one can view the battle accurately at low resolution by focusing on the attacker and defender force levels and force ratio. It suggests implicitly that these determine the loss rates. As Fig. 3.2 shows, however, using results of the “detailed” model for a particular case (25 divisions vs. 5 divisions on a militarily usable frontage of 50 km), the actual relationship between $RLR(t)$ and $F_{tot}(t)$ is complex, and even mysterious. If Lanchester theory applied to the aggregate model as well as the “detailed” model, RLR would be proportional to $1/F_{tot}^2$. Instead, the curve is decidedly ill-behaved, doubling back on itself. The reason is that $RLR(t)$ and $F_{tot}(t)$ are actually both outputs of a more complex simulation. That is, there are “hidden variables” affecting them; they are hidden in that they do not appear in the

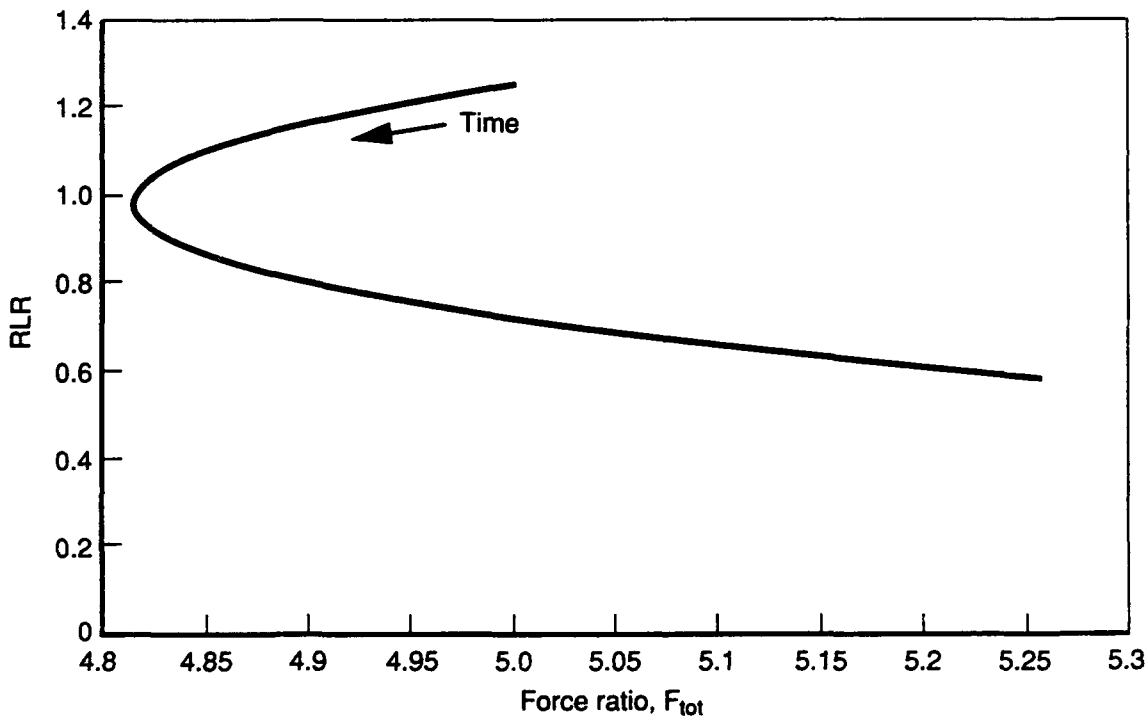


Figure 3.2—Effect of Hidden Variables in Aggregated Display of Higher-Resolution Simulation

aggregated display. In this case, the hidden variables deal with the fraction of each side's forces that can be employed on the FLOT given particular circumstances of terrain and total force levels.

- *Ambiguities when specifying aggregated variables.* The second generic problem is that users viewing things in aggregate often want to *specify* certain variables in aggregate, as part of "What if?" testing. However, there is no way to provide F_{tot} as an input to the underlying simulation. Instead, $F_{tot}(t)$ is an output. To specify F_{tot} , the user must in fact specify both A_{tot} and D_{tot} separately or accept a default disaggregation supplied by an interface model. The interface model's assumptions about how to disaggregate may generate a very different situation than the user would have in mind if he thought about it. In more typical and complex models, the disaggregation problem is much worse.
- *The need for aggregated models.* In cases where the high resolution model is quite complex, users attempting to think and make decisions at the aggregated level also want and need *aggregated models*, not just displays. Why? So that they can fully *comprehend* what they are doing as they reason and consider alternatives, rather than trusting a more detailed model they don't fully understand.⁴ Also, the simpler models are often more appropriate amidst massive uncertainty (the problem of "bounded rationality" discussed in Simon, 1982). And, finally, the simpler models are faster and easier to use when one considers the time necessary to specify assumptions. So it is that many higher-level real-world decisions are analyzed with rules of thumb, back-of-the-envelope calculations, or simple spreadsheet models rather than more complex simulations.⁵

Alternative Submodels

The alternative-model approach is relatively common and is also very useful.⁶ By contrast with selective viewing, this type of variable resolution is indeed *variable-resolution modeling*, not just variable-resolution display. An example might be a model with a complex bomber penetration submodel and, as an alternative, a very simple submodel—perhaps one as simple as specifying survivability and weapon delivery probabilities. Another example

⁴In considering those who like low- and those who like high-resolution combat models, we note that the former distrust the latter and the latter distrust the former. Policy analysts have often seen examples where large and complex models gave wrong answers because of deeply buried assumptions. Analysts who work with high-resolution models routinely have difficulty believing that aggregated models can adequately describe the phenomena.

⁵Wanting a good aggregation does not mean it is easy to get one. Many aggregated models are badly misleading and some good aggregate models are not as intuitive, or even as simple, as one might like. Further, an aggregated model may sometimes be accurate and other times inaccurate (see, for example, Appendix C). Finding and justifying good aggregations is a major objective in theoretical work, but there has been relatively little such work in combat modeling. Instead, aggregations have often been merely postulated.

⁶This is the principal variable-resolution method used in the Air Force model Tac Thunder, the strategic mobility model MIDAS, and the RSAS. There are many other examples. See, for example, the discussion in Bennett, Jones, Bullock, and Davis (1988).

might be a global simulation with: (a) a strategic mobility submodel that optimally allocates units to different types of aircraft and sealift, and then simulates deployment over time, taking into account whether canals are open, convoying is necessary, airfields are available, and so on; and (b) a much simpler submodel in which the user specifies the arrival times of his units after using a utility function to make side calculations based strictly on tonnages, types of units, airlift productivity in ton-miles/day, and rules of thumb for when sealift becomes available. Yet a third example might be a ground-combat model providing the user with alternative ways to compute attrition: (a) heterogeneous Lanchester equations in which unit strengths are maintained for each category of weapons and attrition involves a killer-victim attrition matrix, and (b) a much simpler scalar equation in which attrition is calculated as a function of aggregated force scores, or perhaps a force ratio.

Such flexibility has high payoffs, especially when the time comes for sensitivity analysis, or when one wants to focus on one set of processes while simplifying the treatment of others to reduce complexity or avoid tedious data collection. The low-resolution options are also quite useful when the higher-resolution data simply are not available, but reasonable estimates can be made of the low-resolution inputs.

Despite its usefulness, however, the approach of alternative submodels is a recipe for trouble *unless* there is a heavy emphasis on integration during the design phase. It is all too easy to satisfy conflicting user demands by inserting alternative models and switches without thinking through the overall implications in terms of real or apparent anomalies in observables, or outright validity. Most models advertising variable resolution use the alternative-model approach, but few qualify as being integrated, much less anything one might call seamless.⁷ We emphasize here that a single calibration exercise for an allegedly representative case does not constitute integration.

Integrated Hierarchical Variable Resolution

Integrated hierarchical variable resolution (IHVR) design is design in which certain relevant processes bear well-defined hierarchical relationships to one another. In this case, a lowest-resolution process has as inputs the outputs of higher-resolution processes. However,

⁷In passing we should note that not all families of models need to be integrated, much less designed, for seamless variable resolution. For example, a decision support system developed by RAND to serve the Air Force in managing enlisted personnel numbers and distributions consists of a family of nonintegrated models that deal with very different facets of the general problem. One deals with short-term issues such as achieving proper end strength in a given year and depends on statistical analysis. Another model, based on simulation, deals with multiyear issues and is relevant to Air Force programming. And so on. The models are not closely related, nor should they be, because the problems they deal with are of different classes. In this Note, however, we are concerned with models that "should be" integrated across resolutions.

one then has the option to specify the inputs of the lower-resolution process as data rather than running the higher-resolution processes. We will give examples in the next section.

We argue that the IHVR approach, which provides the interactive capability to change resolutions seamlessly, albeit in steps, is the ideal approach when it is feasible. As mentioned above, by changing resolution "seamlessly" we mean that predictions agree (at least near calibration points) and that the change of resolution is cognitively smooth because of a consistent representation of the system (and a related integrated data base).⁸ By and large, there have been few instances of combat models designed in this way. The approach, however, is one that we believe has considerable merit. Most of the remainder of this Note focuses on IHVR. We give examples in some detail and then discuss some of the difficulties associated with making IHVR work.

Comparisons

Against this background, we can compare the different classes of variable-resolution modeling as shown in Table 3.1, which relies on the attributes discussed at the beginning of the section. The table considers three "types" of simulation: closed, interruptible, and interactive. Closed simulations run to completion once started; interruptible simulations can be stopped partway through a case, at which point parameters can be changed; "interactive" simulations are sometimes synonymous with interruptible ones, but in other usage human intervention is *required*. The next column refers to whether the variable resolution is achieved within a single model or by having a family of separate models. Levels of simulation resolution refers to the underlying model(s), not to displays. Change points are the points at which one can change resolutions.

With these definitions, consider the first case as an example of how to read the table. This involves selected viewing in a closed single model. There is really only one level of simulation resolution, even though one may be seeing displays at various levels. Changes of resolution are possible only in preprocessing (e.g., when examining input data) or in postprocessing—i.e., when examining results. The resolution of inputs must be at the highest resolution treated—i.e., the resolution of the underlying simulation. Zooming can be in or out, but involves only the display. It can be reversed. Results should certainly be consistent across levels of resolution, although there may be some hidden-variable problems as discussed above. Descriptions should also be consistent, with low-resolution displays

⁸It is common for the models of an alternative submodel design to have separate data bases for the low- and high-resolution data. The relationships between the data elements may be quite unclear and there may be only partial consistency, based on occasional and casual efforts to calibrate.

being, in most cases, aggregations of higher-resolution displays. The rationale for the low-resolution results may be obscure, however, because the underlying model deals only with high-resolution phenomena. That is, there is no low-resolution model to explain the low-resolution results.

Upon comparing the different forms variable resolution can take, as described by Table 3.1, the most important distinctions between alternative submodels and IHVR emerge as distinctions in consistency, or degrees of seamlessness. *Some* designs using alternative submodels are clear and consistent, but most are not—and lead instead to alternative submodels having little to do with each other, as manifested in the submodels having unintegrated and even inconsistent data bases. The exceptional designs using alternative submodels often have much in common with IHVR, or may indeed be IHVR designs, even though that may not have been an explicit goal.

Table 3.1
Variants and Attributes of Variable-Resolution Model Types

Type of Variable Resolution	Type of Simulation	Single Model or Family	Levels of Simulation Resolution	Change Points	Resolution of Inputs	Direction of Zooming	Reversibility of Zoom	Consistency of Results Across Levels	Consistency of Description Phenomena	Clarity of Low-Resolution Phenomena
Selected viewing	Closed	Single	1	Pre- and postprocessing	High	In and out (of displays only)	Yes	By definition, but with hidden-variable problems	By definition	Low
Selected viewing	Closed	Single	1	Pre- and postprocessing	Low	In and out (of displays only)	Yes	Usually poor (ad hoc disagreement)	By definition	Low
Selected viewing	Interruptible or interactive	Single	1	Pre- and postprocessing and at interrupts	High	In and out	Yes	High	High	Low
Selected viewing	Interruptible or interactive	Single	1	Pre- and postprocessing and at interrupts	Low	In and out	Yes	Usually poor (ad hoc disagreement)	High	Low
Alternative sub-models	Closed	Single or family	Usually 2, can be more	Pre- and postprocessing at interrupts	High or low	In and out	No	Highly variable (often poor)	Highly variable (often poor)	High

Table 3.1--continued

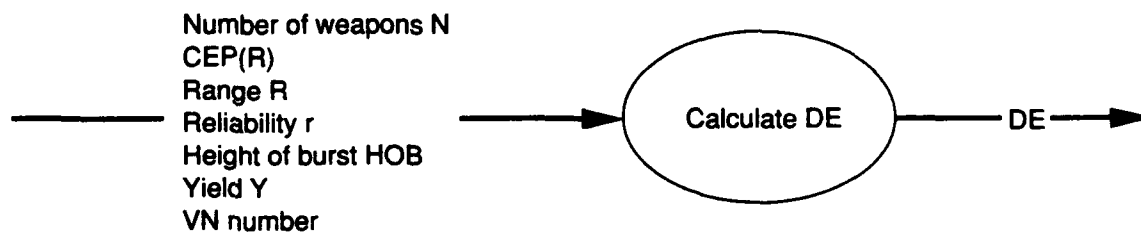
Type of Variable Resolution	Type of Simulation	Single Model or Family	Levels of Simulation Resolution	Change Points	Resolution of Inputs	Direction of Zooming	Reversibility of Zoom	Consistency of Results Across Levels	Consistency of Description	Clarity of Low-Resolution Phenomena
Alternative sub-models	Interruptible or interactive	Single or family	Usually 2, can be more	Pre- and postprocessing and at interrupts	High or low	In and out	Not usually (need a disaggregation model)	Highly variable (often poor)	Highly variable (often poor)	High
IHVR	Closed	Single or family	At least 2, often more	At input time	High or low	In and out	No	High	High	High
IHVR	Interruptible or interactive	Single or family	At least 2, often more	At input time or interrupt points	High or low	In and out	Need a disaggregation model	High	High	High

4. DESIGN OF INTEGRATED HIERARCHICAL VARIABLE-RESOLUTION MODELS

AN ILLUSTRATIVE PROBLEM SHOWING THE VALUE OF IHVR

To illustrate the concept of integrated hierarchical variable-resolution modeling (IHVR), let us first draw on a model that computes the damage expectancy DE when N weapons are fired at a target with a hardness characterized by a "vulnerability number" VN. The weapons are launched from a range R, have a reliability r, a CEP that depends on R, a height of burst HOB, and a yield Y. Figure 4.1 shows the data flow diagram for this trivial model and the equation for computing DE ("a" is a specified constant and F is some specified function of R, VN, Y, and HOB).

To be sure, this model is extremely simple, but we need a simple example to make basic points. Let us now consider how one might use hierarchical principles to design a model providing different levels of resolution. Figure 4.2 shows a data flow diagram for such a model. *Asterisks indicate variables that can be specified initially or interactively by the analyst—i.e., as parameters.* That is, the design allows the analyst to override or bypass certain calculations. The design includes relatively high resolution in one dimension: the weapon's CEP is calculated as a function of the launch vehicle's range from the target,¹ but the functional form for CEP(R) is specified by parameters in the data base. While adding resolution in this respect, the design simultaneously provides some new aggregated modes of operation. The analyst can specify SSPK, thereby working at a rather high level of aggregation; alternatively, he can specify CEP and target hardness in p.s.i., thereby working



$$DE = 1 - \{ \text{rexp}[-aF(R, VN, HOB) / CEP(R)^2] \}^N$$

Figure 4.1—A Model for Calculating Damage Expectancy

¹This is significant when dealing with sea-launched ballistic missiles (SLBMs) or aircraft, since there are tradeoffs between standoff range and survivability on the one hand, and range and accuracy on the other.

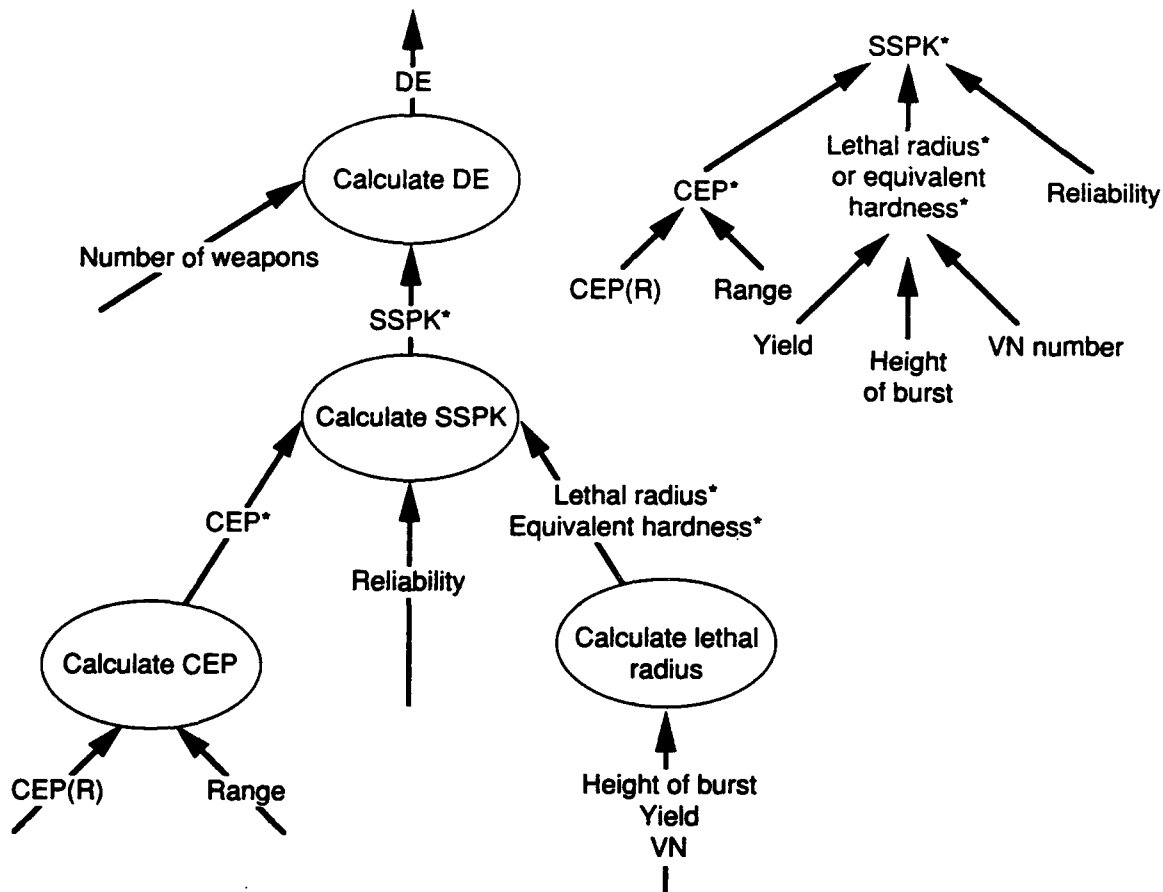


Figure 4.2—A Damage-Expectancy Model with Integrated Hierarchical Variable Resolution

at a somewhat lower level (higher resolution); or he can let SSPK and DE be calculated using all the low-level processes dependent on weapon range, VN number, and so on. He also has options with respect to specifying or calculating lethal radius.

As the tree-like structure on the right side indicates (this is an alternative depiction of what is going on, up to the level of SSPK, not part of the data flow diagram itself), the design involves a strict hierarchy of variables, with each variable in the tree affecting only the variables above it. Because of this, there are no side effects when the analyst chooses to work at a higher level of aggregation (more on this below). It is not always possible to work in such hierarchies, but doing so is highly desirable if one has a choice. Why? Because zooming *can* be seamless.² Results can be consistent across levels of resolution (at least near

²The use of "can be" is significant here. Merely because one uses hierarchical design methods does not imply that the results will be valid or understandable.

calibration points), although, of course, high resolution will generate more information than low resolution. And, importantly, the *concepts* and *worldview* can be the same across levels of resolution. There is no cognitive disconnect when one moves up or down the tree in Fig. 4.2, because the concepts are all consistent and explicitly linked in a natural way.³ In more complex examples, with trees having many levels, it is straightforward to have multiple alternative levels of resolution: One merely decides at what level to truncate the hierarchy.

One aspect to this type of seamlessness is that one knows from the diagram, in theory at least, how to establish baseline values of the parameters one might specify directly in preference to running the more detailed calculations. For example, a baseline value of CEP might be determined by the following calibration equation:

$$CEP_{base} = \sum_i \int_{t_1}^{t_2} \int CEP(R) P_i(R, t) dR dt$$

This evaluates an average value of CEP over a period of simulated time across an appropriately weighted set of scenarios. That is, for scenario *i* and time *t* one would find the average CEP by integrating the product $CEP(R)P_i(R(t))$ over all ranges *R*, where *P* is the probability density for finding a launcher (SLBM or bomber) at a range *R*. That, in turn, depends on the scenario and time, so one must integrate over time and sum over scenarios as well.⁴

Figure 4.3 illustrates with a caricature of a dialog box how a designer might want a user to be able to specify which mode he works in. The example shows him having decided to specify CEP and Lethal Radius as parameter values, thereby bypassing the parts of the model that calculate those variables. Alternatively, he might have specified a predefined "mode" (by typing in a name or number), which would then be interpreted as implying particular settings of Yes or No for the SSPK, CEP, and Lethal Radius boxes.

³Alternative submodels can be and sometimes are integrated in the sense of mappings and calibration mechanisms being well defined and documented. It is more difficult for the integration to be easy to understand and follow at a glance. Indeed, when one tries to integrate alternative submodels to achieve this, the tendency is to produce something very much like, or identical with, IHVR.

⁴This type of calibration equation is not always appropriate. If, for example, there were only two ranges of interest, one very short and one very long, the CEP for an "average range" would relate to neither case. This is merely one of many examples in which expected-value calculations must be scrutinized.

Specify mode	Specify CEP?	Specify lethal radius	Specify SSPK?
	☒	☒	☐
☐			

Figure 4.3—A Dialog Box for Specifying Level of Resolution

At the level of computer code, this design might be implemented with statements such as:

```
If user-specified-SSPK is None
Then Perform Calculate-SSPK.
Else Let SSPK=user-specified-SSPK.
```

```
If user-specified-CEP is None . . .
```

In this example based on the RAND-ABEL™ programming language,⁵ “Perform” is a key word invoking a function. Thus, Calculate-SSPK is the function referred to in the bubble diagram of Fig. 4.2. “User-specified-SSPK” is a parameter that may have no value specified or a value to be used.⁶

In our experience, programmers would be unlikely to implement a DE model in the way indicated in Figs. 4.2-4.3 unless they were sensitive to variable-resolution issues from the outset. Further, asking them to add a low-resolution option after the higher-resolution program existed (e.g., asking to add the option to be able to specify SSPK directly) might lead to problems, especially if there were time pressures. For example, the quick patch might produce the correct DE, but there might still be displays showing the yield, CEP, height of burst, VN number, and calculated value of SSPK even though the SSPK value used

⁵See Shapiro, Hall, Anderson, LaCasse, Gillogly, and Weissler (1988) and Davis (1990).

⁶In the extreme, the program would be designed so that each parameter was a special case of a function that could be more complex (and that would have its own parameters), down to some level of detail. That would provide flexibility, but introduce a great deal of initial overhead. It might also reduce comprehensibility. We speculate, however, that it should be possible to generate automatically the code providing the option to parameterize (i.e., code analogous to that above). That is, one would specify on a graphical interface the functions that one wanted to be able to bypass (by specifying parameters), and a utility program would generate the relevant code.

in the DE calculation would be the one specified directly by the analyst and would probably be inconsistent with the detailed variables.

If the analyst made his requests for flexibility incrementally, the result would probably be a series of patches that would be hard to verify, understand, and maintain. Suppose that the analyst first asked to have CEP calculated as a function of range; then, a week later, asked to have the option of specifying CEP directly reinstated; and then, a week later, asked to be able to specify target hardness in p.s.i. rather than in VN number. It is unlikely that the resulting code would end up reflecting a coherent variable-resolution concept such as that illustrated in Figs. 4.2 and 4.3. The conclusion here is that

- Prior design pays major dividends when one wants variable-resolution capabilities.
- Designers should anticipate later requests for more (or less) detail and should leave appropriate stubs, even if that temporarily adds program complexity.

No reader should interpret this to mean that we are suggesting that everything can be anticipated or that designs should be so complex as to hinder creative exploration and rapid prototyping. However, there is a balance to be struck between up-front designing and learning from doing. If we seek variable-resolution models with minimal inconsistencies, the balance should be pushed farther toward initial design. As new modeling environments and programming languages enter the scene, the tradeoff may be less troublesome, since later revisions of code will be less painful.

COMPLICATIONS IN DESIGNING FOR HIERARCHICAL VARIABLE RESOLUTION

Having discussed an example of idealized integrated hierarchical variable resolution, let us now review some of the difficulties that arise in attempting to achieve it in practice. In particular, let us consider: (a) side effects, (b) the distribution of conceptual models throughout a computer program, (c) imbedding low-resolution models in existing higher-resolution models, (d) synchronization, (e) branching vs. override issues, and (f) lack of appropriate software design tools. This list is by no means comprehensive, but it reflects the range of problems to which we are sensitive by virtue of our personal experiences and observations. Some of these problems apply also to using the alternative submodel approach to variable resolution. Indeed, some of them relate to the more general problem of achieving appropriate *modularity*.⁷

⁷For a detailed discussion of hierarchical modular design and its virtues, see Zeigler (1990). His discussion focuses on hierarchies in objects rather than hierarchies in processes, but his emphasis on rigorous modularity is quite relevant here.

Side Effects

First, let us consider side effects, using a concrete example and then a more generic depiction. Figure 4.4 assumes a nuclear-weapon simulation that computes damage to hard targets and collateral fatalities from fallout. For simplicity it assumes that all targets are the same and all weapons are the same. Whereas in Fig. 4.2 there was a neat hierarchy of variables that could be truncated at any level, here there is *cross talk* between processes: Certain variables (yield and height of burst) affect higher-level variables in two different trees. Suppose that a user specifies the SSPK directly as 0.1 for the sake of a parametric exploration, thereby bypassing the calculation of fatalities that makes use of yield, height of burst, and so on. Suppose, however, that the value of yield is 10 MT and, upon looking at displays of the fatality calculations, one of the user's colleagues notices high fatalities because of the high yield. He might then note the low SSPK being reported and say to himself, "How can that be? With such a huge yield, the SSPK really should be larger than 0.1. I know that the target isn't *that* hard. There's an inconsistency here." In some cases, he could simply ask his colleague and have the inconsistency explained, but in other cases there would be a problem of real or apparent inconsistencies. This is an example of an unintended side effect of the design.

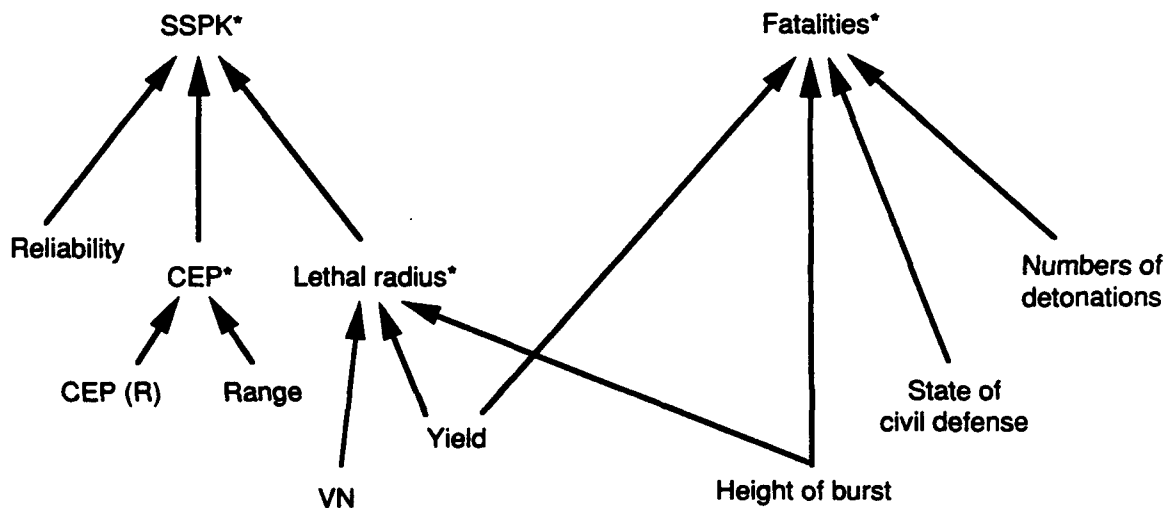


Figure 4.4—An Example of Nonhierarchical Relationships

How does one avoid such problems? There are several options: (1) tolerate the inconsistencies;⁸ (2) create self-consistent modes of operation in which an analyst, if he wants to specify SSPK directly, must also specify something like the fatalities per detonated weapon directly so that yield and height of burst would disappear from the simulation entirely; or (3) to cover cases in which the analyst specifies SSPK, have a process to estimate the fatalities per detonated weapon in some way that doesn't depend on yield and height of burst,⁹ and then create a corresponding display (since the user would not see yield and height of burst, there would be no observable inconsistencies). Solution 3 would demand less of the analyst, but it might necessitate some crude approximations.

Figures 4.5 to 4.6 now show an illustrative variable-resolution design that would avoid the problems of inconsistency seen in the previous example. In the higher-resolution mode (Fig. 4.5), SSPK and fatalities would be computed, as before, from low-level processes with cross talk among trees. The variable height-of-burst policy would not be used (hence the dashed line). If the analyst chose to work at a higher aggregation (lower resolution, in

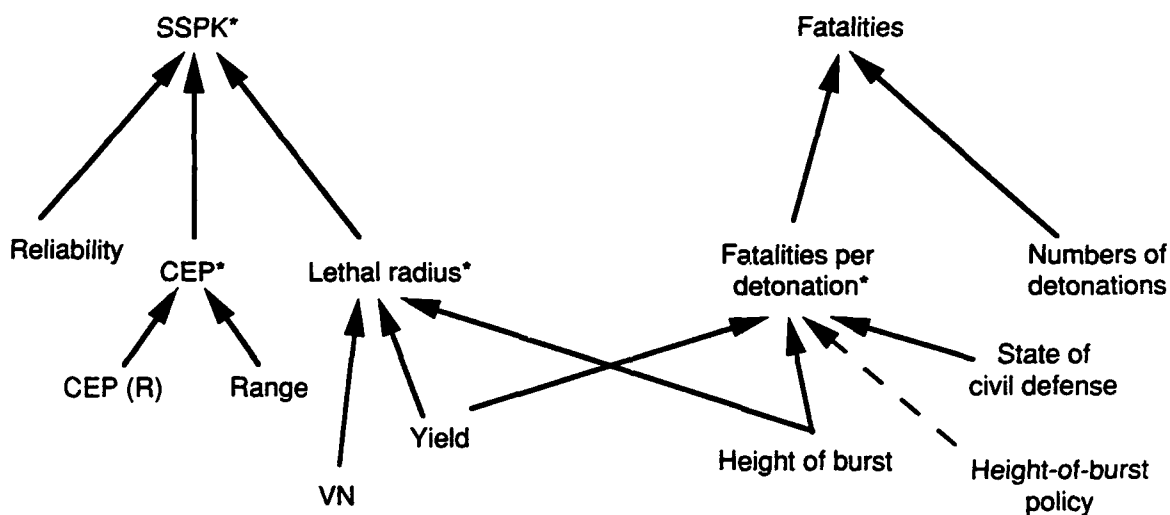


Figure 4.5—A Revised Design Viewed for the High-Resolution Case

⁸This is often the method of choice within the context of a study, because the users are aware of the inconsistencies and can manage them. The story is different when the model is to be used externally.

⁹The modeler might note that many previous studies have concluded that a counterforce attack with 2000 weapons would kill approximately 20,000,000 people. He might then insert a "model" in which the fatalities per detonated weapon would be 10,000. Obviously, the model would not be sensitive to targeting choices such as air bursts vs. ground bursts, but there would be no apparent inconsistencies.

Fig. 4.6) in which he could specify lethal radius or SSPK directly, then he would also be required to specify either fatalities per detonation or height-of-burst policy.

The point here is that because of side-effect problems one cannot go about creating variable resolution arbitrarily unless one is willing to accept inconsistencies, but if one eliminates inconsistencies one may be forced also to reduce the resolution (and perhaps the quality) of certain other submodels.¹⁰

Figure 4.7 illustrates the issue of side effects more generically and uses an example of a dynamic model such as one would find in a simulation. Figure 4.7 is a data flow diagram for computations within a given time step (Δt) of a simple model. Let us suppose that the intended low-resolution mode focuses on X_3 , X_4 , d_3 , and d_4 , where the d 's are static data elements and the X 's are dynamic variables. The higher-resolution picture involves additional variables X_1 and X_2 , along with related static data elements d_1 and d_2 . Each of the variables and data elements may be regarded as a vector for greater generality. Thus, by d_1 we really mean the set of static data items that are used as part of the process computing changes in state-variable X_1 (or, simply, variable X_1). And by X_1 we mean a set of variables $X_{11}, X_{12}, \dots, X_{1N}$.

Let us now consider what happens when we try to achieve low resolution by truncation. Perhaps the analyst knows that X_1 doesn't usually change much in the real-world processes he is interested in, or that it changes in only a very simple way. He may

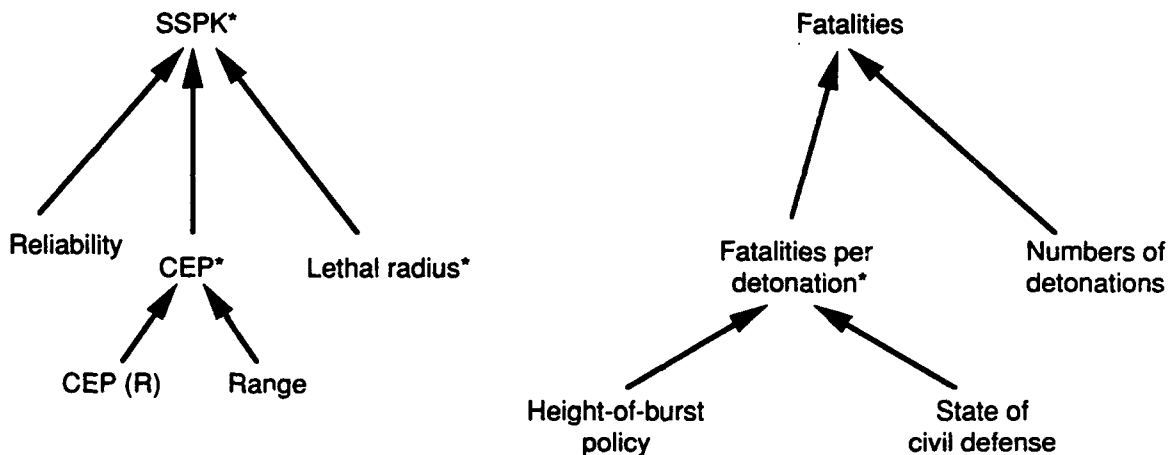


Figure 4.6—A Revised Design Viewed for the Low-Resolution Case

¹⁰Side-effect problems are, of course, important in software engineering generally, not merely in instances of variable-resolution design.

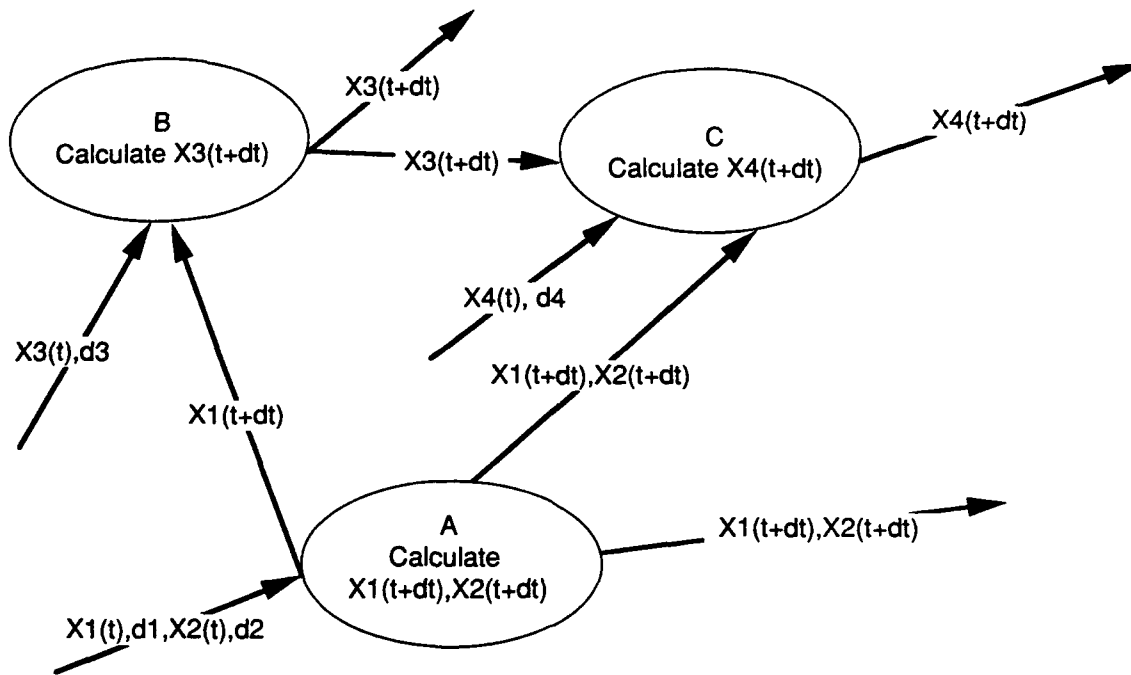


Figure 4.7—Generic Example of Side-Effect Issues in Variable-Resolution Modeling

then wish to specify $X1(t+dt)$ with an alternative process A' consisting of some very simple calculation dependent only on parameters (e.g., a half-life for an exponential decay). If the programmer tries to implement this "specification," however, he will (or should) find a problem: What should he do about $X2$? Although the analyst was interested only in simplifying the computation generating $X1(t+dt)$, it turns out that the module generating $X1(t+dt)$ also generates $X2(t+dt)$, which is a necessary input of module C. Furthermore, while the analyst may have felt that it was acceptable to approximate $X1(t+dt)$ with the simple process A' , his judgments on this may have been dictated by the relative insensitivity of $X3$ to the precise value of $X1$. But process C generating $X4(t+dt)$ may be more sensitive to $X1$, and there may be still other processes elsewhere in the model that depend on $X1$ as well. So, what happens if we truncate the simulation by omitting Process A? The implementer must: (a) use the same value of $X1(t+dt)$ for both Processes B and C, and for any other processes that use $X1(t)$, and (b) decide on his own what to do about $X2$ (e.g., make $X2$ into a constant parameter or develop some simplified algorithm for $X2(t+dt)$, in which case he would be substituting that algorithm for the more complex A process). The implementer is therefore determining what the *real* model is, and the analyst may be proceeding without even knowing that his specification was incomplete.

In a worst case, the implementer might not even notice the "other" output of Process A, and the simulation might fail because Process C, expecting a value for $X_2(t+dt)$, wouldn't find it. Or, in a more likely case, the system would run on, but Process C would be using an incorrect value of $X_2(t+dt)$ coming from some other process not shown in Fig. 4.7 as an input into Process A.

This simple example brings out an important principle: To specify an aggregation or simplification (or even a change), one must specify how *all* the outputs of bypassed or omitted processes are now to be computed or what assumptions are made to eliminate them. One cannot focus solely on the variables that happen to be of interest to the particular analyst's problem. An important observation at this point is that *current software tools do not generally provide conveniently the information needed to avoid these problems.*¹¹ We shall have more to say on this below.

Distributed Effects

Often, the modules of an analyst's conceptual model have little in common with the modules of the more general model as implemented in a computer program. The most obvious example here is attrition. An analyst may think of ground-unit attrition or bomber attrition as natural "chunks," but the implemented model may be organized so that the various sources of attrition are distributed throughout the code—i.e., are calculated in widely separated modules (e.g., separate modules for attrition to ground units due to other ground units, close air support sorties, helicopter sorties, or nuclear blasts). This means that if the analyst proposes a more aggregated model of attrition (or even a different model of attrition), implementing the changes will require changes in multiple elements of the program (Fig. 4.8). Further, it may or may not be the case that the program modules affected deal only with the subject at hand (e.g., ground-unit attrition). And, to make things even worse, the various affected program modules may have been written by different programmers using somewhat different styles. All of this makes for significant complications. It also encourages a tendency to patch one part of a problem without implementing a comprehensive change. That, in turn, tends to create side effects and outright errors.

¹¹Some software environments are much better than others, of course. As a point of comparison, it is painful in a C/Unix environment with complex models using pointers and structures to get good summary information on the inputs and outputs of all key functions. By contrast, with some higher-level strongly typed languages (e.g., the RAND-ABEL language, which operates within a C/Unix environment), it is fairly straightforward to do so because the language includes a data dictionary and what is called a cross-referencing tool. Ironically, the much "simpler" spreadsheet language, Microsoft's EXCEL™, also has such tools, but most general-purpose programming languages and environments do not.

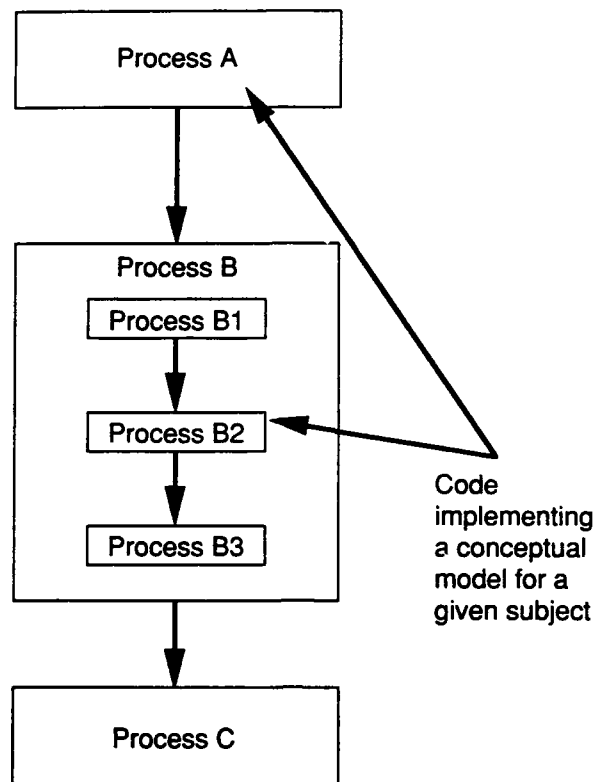


Figure 4.8—Example of a Distributed Model

Imbedding a Low-Resolution Submodel Within an Existing Higher-Resolution Model

Another generic problem is that when one attempts to add a low-resolution submodel to an existing higher-resolution model, as a variable-resolution option, the analyst or modeler designing the addition often does not fully understand the context in which it is to operate or the information that must be provided in the specification. Suppose the existing model includes a simulation of attacks on bomber bases by sea-launched ballistic missiles (SLBMs) launched from points relatively near the coastline. The number of bombers surviving this attack in a given simulation run will depend on the number of SLBMs, their launch locations, the trajectory of the missiles, the yield of the weapons, the laydown pattern of weapons, the number of bombers on the airfield, the number of runways, alert state, warning time, and bomber "flyout" characteristics and tactics. All of this constitutes a fairly complex problem in which some fraction of the alert bombers manages to escape their bases before the bases are destroyed. This fraction is called the prelaunch survivability (PLS), although the name is not quite appropriate. Consider now an analyst proposing to specify PLS as a parameter in a more aggregated depiction of bomber operations. Or, perhaps he proposes to

calculate PLS in a much simpler way. Chances are, he will provide closed-form analytic equations or a simple algorithm and believe he has specified the model. In fact, however, if the model is to be imbedded in a larger simulation model, the model is not really specified until the analyst indicates precisely where and how, in the actual control flow of the extant program, the notions of the original model are to be reflected.

Often in such work the analyst is not really thinking in terms of simulation when he develops his model. As a result, there is considerable ambiguity about issues such as where the necessary inputs will come from (are they already state variables, and if so where are they updated?). Also, in thinking about something like bomber PLS, the analyst leaves ambiguous matters such as "How should bombers be considered to change state, and when?" Even the concept of "state" is associated with simulation rather than a closed-form analytic model or parameter. Should all the attrition be charged to ground-based bombers with no state corresponding to bombers that are in the process of escaping their bases? If so, what happens if one tries to use a PLS model in the midst of a simulation in which there are already bombers in the escaping state? Or, if one parcels out the attrition implied by PLS over bombers in both ground-based and escaping states, how does one do so? And so on. Figure 4.9 sketches out the situation in quasi code, with the column on the left indicating the possible structure of low-level (high-resolution) code, and the right column indicating the possible structure of a lower-resolution model that is being suggested, one that would substitute a single parameter (PLS) for the processes that otherwise would be used to calculate prelaunch survivability. As the figure illustrates, the "simple model" is not completely specified, because it says nothing about what happens to bombers that are already in the process of launching, or that are on airborne alert, etc. Probably, the analyst wanting to use the lower-resolution model doesn't want even to recognize states such as "launching," but if the option to use the lower-resolution PLS model is to be available interactively, then the designer has no choice: He must specify what happens to aircraft in the "other" states.

Synchronization

Yet another problem when attempting to relate models of different resolution (and in other cases) is that the models may differ in how they deal with time. One may have a more coarse-grained description than another (e.g., updating state every 6 hours rather than every 4 hours), and yet another may be event-oriented rather than time-oriented. A recent DARPA-sponsored project at the MITRE Corporation has demonstrated a powerful set of

Low-Level Model	High-Level (Low-Resolution) Model
[Change attributes of bombers in each of the permitted bomber states]	[Change attributes of bombers in each of the permitted bomber states]
For nonalert bombers, Calculate attrition due to weapon arrival Calculate attrition due to sabotage React to orders changing state	For nonalert bombers, Calculate attrition due to weapon arrival (killing the fraction (1-PLS), where PLS is pre-launch survivability) React to orders changing state
For alert bombers on ground, Calculate attrition due to weapon arrival Calculate attrition due to sabotage React to orders changing state	For alert bombers on ground, Calculate attrition due to weapon arrival (killing the fraction (1-PLS)) React to orders changing state
For bombers in act of launching, Calculate attrition due to weapon arrival on base and escape corridor Calculate attrition due to sabotage React to orders changing state	
For bombers on airborne alert, Calculate attrition due to loiter-area barrage React to orders changing state	
For executed bombers, Calculate attrition due to defenses Attack targets Calculate attrition due to defenses in egress	For executed bombers, Calculate attrition due to defenses Attack targets Calculate attrition due to defenses in egress
Etc.	Etc.

Fig. 4.9—Comparison of Quasi Code for Low-Level and Incompletely Specified Higher-Level (Low-Resolution) Models

procedures and semi-generic software for permitting the interoperability of dissimilar simulations. It depends on reformulating the concepts of the constituent models in object-oriented discrete-event-simulation terms so that the coordination techniques can be applied (Weatherly, Seidel, and Weissman, 1991). These allow the models to operate together, but by no means assure that the results make sense.

Branching vs. Overriding

Another problem that can arise in designing and implementing variable-resolution models is that there may be misunderstandings about whether one is providing the capability to override a part of a model's output temporarily, to override that output permanently (until one intervenes again), or to substitute one model for another (compare Figs. 4.10 and 4.11). In the first instance, the model continues to run and when next it executes it will override the earlier override. In the second instance it will continue to run,

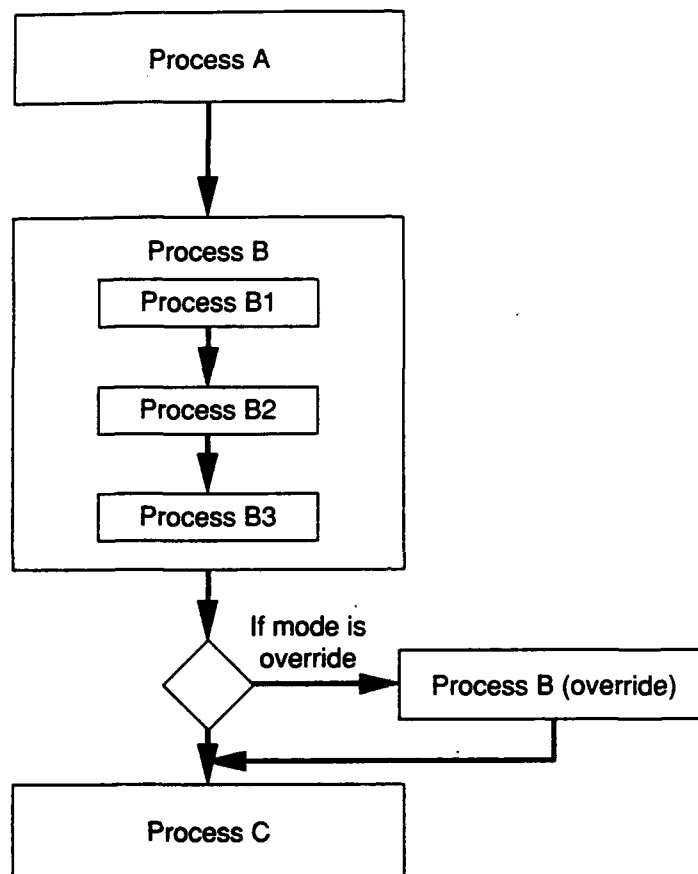


Figure 4.10—Control Flow for an Override Option

but the override will be repeated automatically. In the third instance the submodel is replaced altogether (Fig. 4.11).¹²

Lack of Appropriate Software Tools

This is not the right place in which to discuss software tools in detail, but it is quite appropriate for us to observe here that many of the generic difficulties we have just discussed would be greatly mitigated with better software tools. In the past, however, combat models have not been developed in software environments containing such tools. We anticipate

¹²This and some of the other examples may be regarded as merely examples of poor design, but our point is that designs are *often* such as to make variable-resolution adaptations difficult, and that there are many fairly common classes of difficulty. The root issue is that the models were not originally designed with variable resolution in mind. The flow in Fig. 4.10 could readily arise because of a well-intentioned "patch" that failed to recognize the choices shown in Fig. 4.10 vs. 4.11.

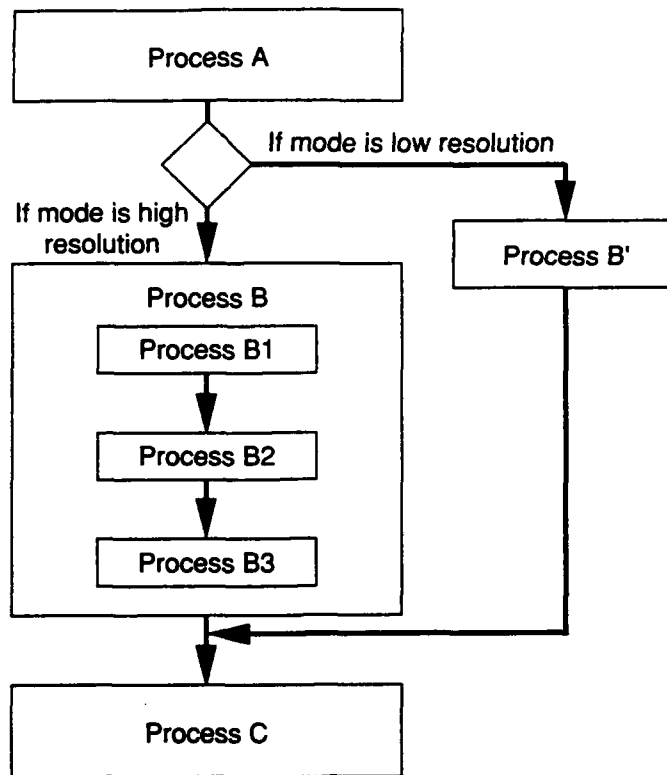


Figure 4.11—Control Flow for a Bypass (Substitution) Option

major improvements in this situation in the years ahead. Object-oriented modeling and programming is useful, because it forces modularity and encourages variable-resolution design of objects (i.e., entities). Advanced high-level languages with data dictionaries and “English-like syntaxes” can greatly reduce the barriers among analysts, modelers, and programmers. Also, there is great value in moving toward graphical specification and documentation methods, including languages. Control-flow, data-flow, and state-transition diagrams are all quite useful, especially if developed in a top-down manner. They can significantly reduce problems of the sort we have discussed in the preceding subsections. An important obstacle to be overcome generally is the widespread belief by programmers that it is right and proper that there be major differences between the structure of programs and the structure of the conceptual model. In our experience, if implemented models are to be comprehensible, especially with variable-resolution features, it is essential that the

differences between conceptual model and program be minimized. Again, then, higher-level specification techniques are highly desirable.¹³

OTHER ISSUES IN CONSIDERING IHVR

The preceding subsections have illustrated IHVR design and have dealt with technical problems implementing it (and, in some instances, alternative submodels). Here let us consider some higher-level issues: whether the real world is hierarchical (if it is not, then one must ask whether hierarchical design is appropriate), whether IHVR unduly limits one's flexibility by imposing a perspective, and whether there are examples of IHVR that represent proofs of principle.

Is the Real World Hierarchical?

The essence of IHVR is, of course, the use of hierarchies of processes. It may reasonably be asked whether one can expect such hierarchies to be natural representations of the real world. In our experience to date, there are indeed *many* problems that can be treated hierarchically (see also Simon, 1982, in which a Nobel Prize winner discusses the importance of nearly hierarchical structure in a myriad of natural systems). The word "nearly" is important here, because in many cases of interest, and in the modeling of air-land battle in particular, the interconnectedness of phenomena creates difficulties and achieving fully hierarchical models is impossible without approximations (e.g., assuming that certain variables are only slowly varying, and can therefore be treated as constants over time periods of hours or even a day). Exceptions must be made to the strict hierarchical design. We discussed in the previous section how approximations can sometimes be made to break cross-relationships between branches, thereby creating hierarchies. Another problem, however, is that some variables are used by many other variables in a variety of different "modules," even modules at different levels of detail. Some of these can readily be treated as global variables outside the hierarchical concept. For example, time and the geographic boundaries and locations of key entities may be important at several levels of a hierarchical model.

More troublesome are variables used in *and affected by* diverse portions of a model. They should be minimized if possible. When they cannot be avoided, however, special care

¹³See Zeigler (1990) for a good description of modular hierarchical design that includes object-oriented concepts and a top-level design concept called system entity diagrams. See Rumbaugh, Blaha, Premerlani, Eddy, and Lorensen (1991) for an excellent discussion of object-oriented modeling that includes extensive use of diagrams. See Davis (1988b) for an example of an IHVR design implemented in a language and environment allowing close coupling between conceptual model, program, and interface. See Rothenberg (1990) for discussion of rapid prototyping and modeling.

should be taken to maintain their visibility and keep them under control. Otherwise, there will be significant side effects of the types discussed in the previous section.

The Problem of Perspective

Achieving variable-resolution design requires that one make many choices about the perspective to present. Even at a given level of resolution there may be several perspectives reflecting differences in the way different people view a problem. For example, one person might think of specifying a bomber's nominal range and nominal payload as attributes, and providing a function to compute actual range for an actual payload and flight strategy. Another person might think of specifying a bomber's flight path as an attribute, allowing that attribute to take on such values as high-high-high, high-low-high, or high-low-low. There might be a table of data translating this attribute into range and payload.

The point here is that if variable resolution depends on the existence of natural hierarchical levels, then we must recognize that different hierarchies may be equally valid and equally natural. Given a particular hierarchical design, one will not have some alternative levels of resolution available from a different hierarchical perspective—at least, not without modifying the model itself.¹⁴

Figure 4.12 illustrates the point by showing variable relationships in two hierarchically designed models at the same level of resolution. The one on the left keeps air and ground forces separate when estimating attrition; the other combines them into a firepower score. Superficially, one might think there is more of a relationship between the two depictions than there is. For example, both have something called effective force ratio, but they mean different things.

The issue of perspective is very important and must be addressed head-on in initial design. One can, of course, change hierarchies later. Nonetheless, it is highly desirable to "get them right" in the first place, even if that means a longer design period before rapid prototyping experiments begin.

Existence Proofs

To prove that IHVR *can* work, we need only a successful example or two. In fact, there are such examples, although the IHVR approach has not, to our knowledge, been widely considered or applied in the past. Let us describe some examples briefly.

¹⁴This problem can be mitigated, however, by using selective viewing techniques to provide information from perspectives different from that reflected in the basic IHVR design.

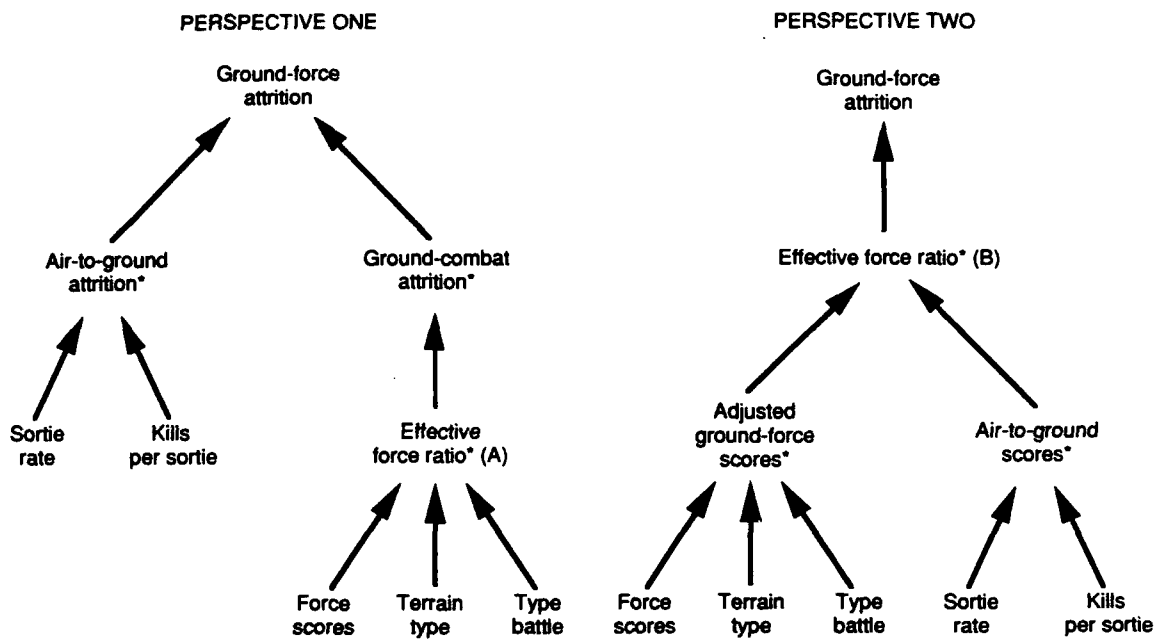


Figure 4.12—Alternative Perspectives Reflected in Hierarchical Designs

Political Models. The first example, which is the earliest one of which we happen to be aware, involved Red and Blue political-level decision models in the RSAS, models developed during the Cold War to reason about issues such as whether to escalate the level of conflict or accept various conditions of termination. These models¹⁵ were designed from the outset with multiple objectives that included (a) using the models within global war games (in a sense “generating scenarios,” but also reacting plausibly to events and the decisions of other models or human teams); and (b) providing a capacity for *analyzing* the likely success of alternative strategies for deterrence, coercion, and war termination. For purpose (a), the models needed to use details of the simulation’s current state. For purpose (b), more aggregated descriptions were needed because, ultimately, the top-level decisions depend naturally on aggregated variables, not details. The design used hierarchical principles extensively.

Figure 4.13 illustrates one such hierarchy from that work, one producing a high-level decision concept called theater-status (i.e., a measure of “How are we doing?”). The actual

¹⁵See, e.g., Davis (1987) for an overview of objectives and methods; see Davis, Bankes, and Kahan (1986) for the initial high-level design.

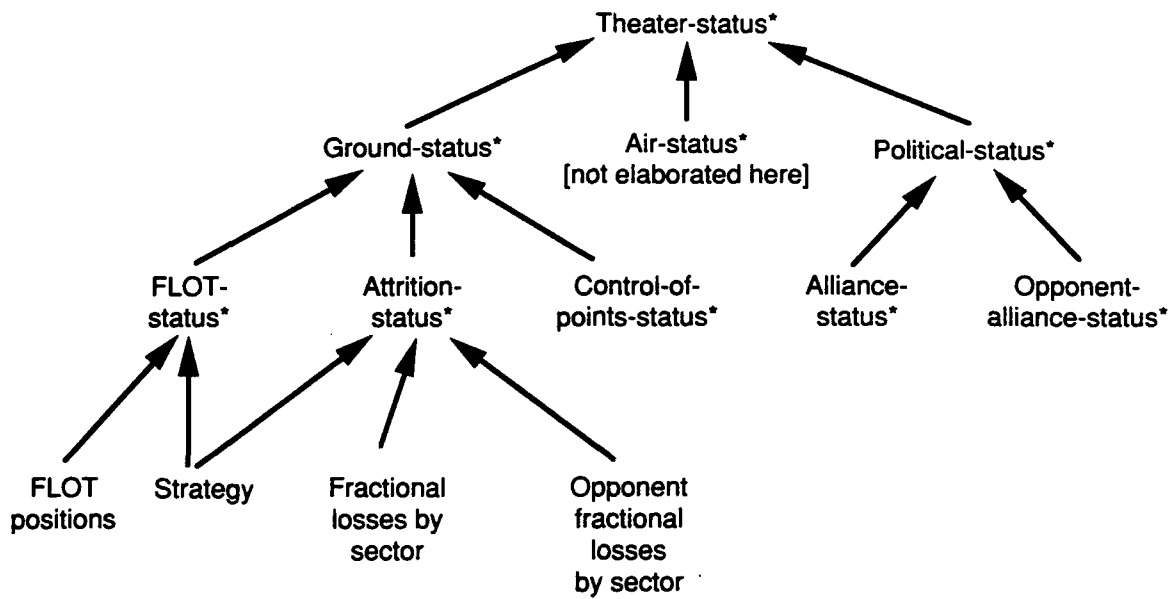


Figure 4.13—Example of Political-Model Variable Resolution

models are large and complex (about 15,000 lines of a high-level language, RAND-ABEL) and contain many such hierarchies. Thus, decisions about escalation and the like depended on variables such as strategic-warning, tactical-warning-of-escalation, theater-status, model-of-opponent, and so forth. Each of these high-level variables could be specified by the analyst or could be evaluated with knowledge-based models at alternative levels of detail. In the extreme, the input to those models came from a combat simulation of global conflict, one generating FLOT traces, attrition, and so on, by theater and sector. As in earlier examples, asterisks in Fig. 4.13 indicate variables that could be set directly by the analyst. These models proved highly flexible and, because of the general modeling and programming approach, relatively easy to understand and modify. Importantly, the original hierarchies stood up well over time, despite massive changes in the details (i.e., in rules within individual modules). The initial design took weeks, not months (although it reflected considerable domain knowledge).¹⁶

Command-Control. Another example of implemented IHVR involves hierarchical treatment of military commands, each of which is represented by a decision model, which is

¹⁶The same techniques used here for political modeling are also quite useful in dealing with a variety of "soft factors" and behavioral models important in command and control modeling at the tactical and weapon-system level. For discussion of some of these factors, see MORS (1989) and Davis (1989).

in turn driven by hierarchical logic. Figure 4.14 indicates schematically the hierarchy of variables by which a decision model representing a theater commander might decide on operational strategy. This approach was used extensively within the RSAS through 1989, especially for Central Europe, before the strategic landscape changed, rendering obsolete the decision models that had been developed for this purpose. As Fig. 4.14 suggests, the analyst could specify the theater-level operational strategy directly as a qualitative "parameter" (e.g., forward defense vs. forward defense in CENTAG and defense at the Weser River in NORTHAG). Alternatively, a more complex knowledge-based logic would be used to decide the operational strategy as a function of inputs such as political-level guidance coming from a political model (that mentioned in connection with Fig. 4.13) or the analyst and numerous situational variables such as force ratios, densities, movement rates, and estimated times before breakthroughs, which were produced by the combat-model part of the overall simulation.¹⁷ Here again, the intention was to provide options for both fully integrated closed simulation and more tightly controlled analysis.

We count this as a second example of IHVR even though it is also an RSAS example, because the military decision models are of a very different nature than the political models, and were built by different individuals.

Theater-Level Decision Aids. As our final example here, Fig. 4.15 shows some of the variable-resolution features of the Generalized Force Ratio Model (GEFRAM) developed

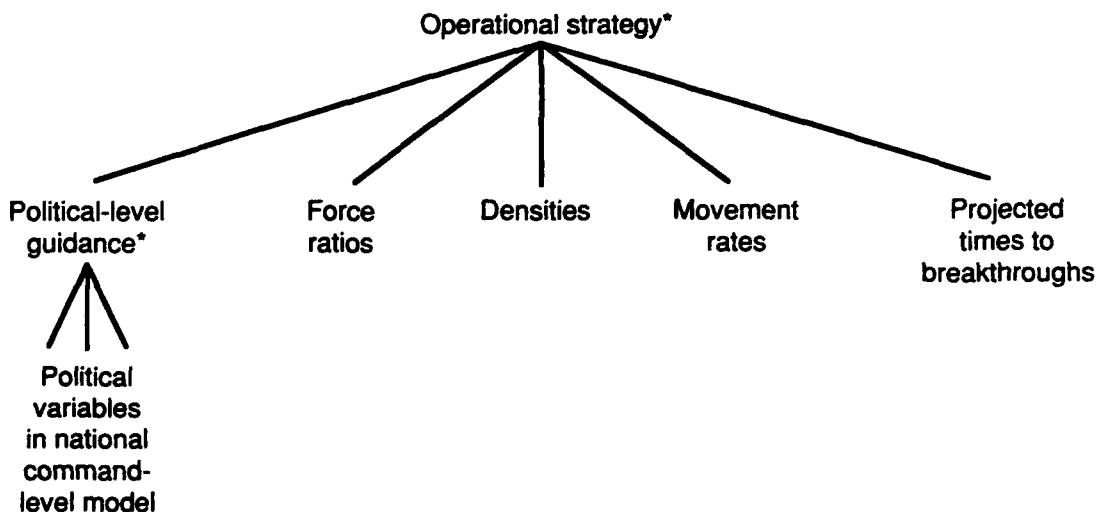


Figure 4.14—Example of Variable Resolution in Command-Control

¹⁷For details on these military decision models see Schwabe and Wilson (1990). For a more philosophical treatment, see Davis and Howe (1989).

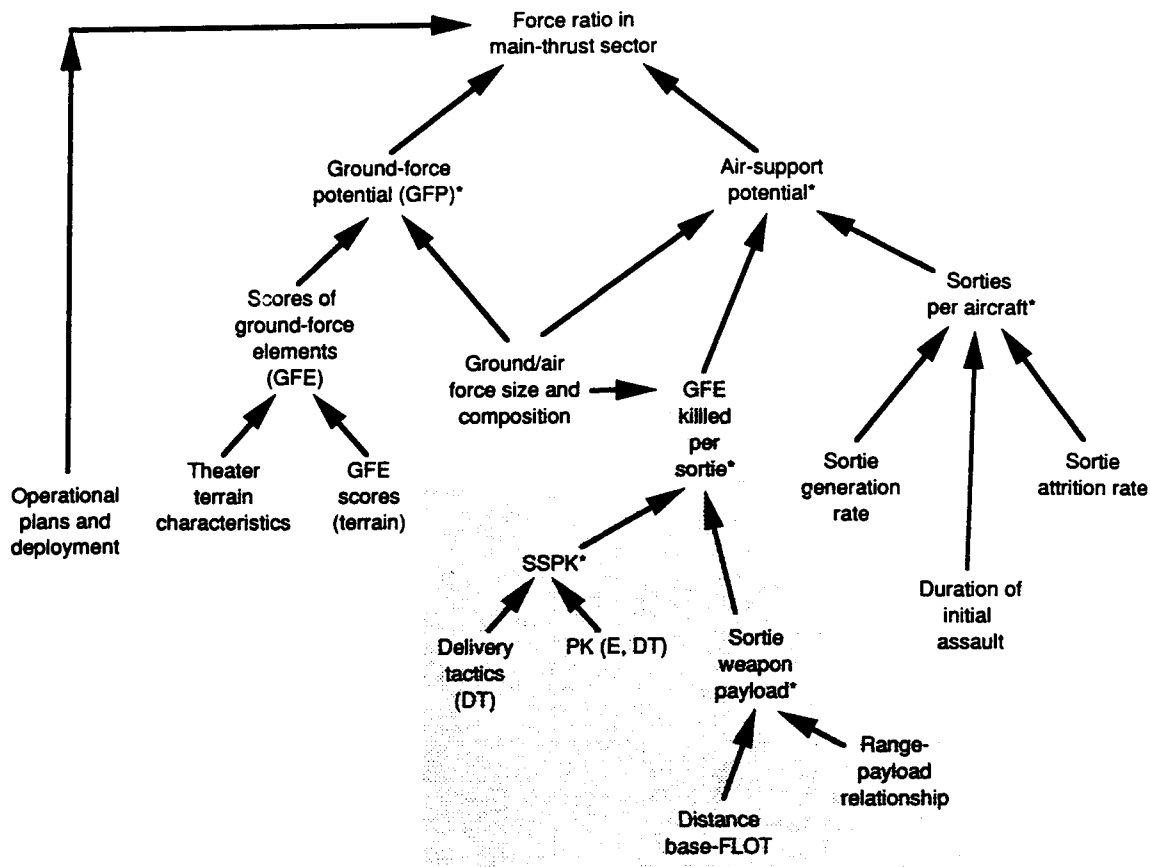


Figure 4.15—Variable Resolution in the Generalized Force Ratio Model

by Huber (1990b) for the purpose of estimating the attacker-to-defender force ratios expected in the main thrust sectors of a theater, given the available forces and the operational plans and objectives of attacker and defender. The asterisks again indicate the variables that may be directly specified by the user, thus cancelling their calculation in case the user feels more competent to estimate the respective lower-resolution parameter values. For example, when assessing the impact of tactical air on the main thrust sector force ratio, the user may want to calculate the local air support potential based on his estimate of the sortie effectiveness in terms of ground-force elements (GFE such as tanks, infantry fighting vehicles, artillery pieces, dismounted infantry units, etc.) killed per sortie over target. Alternatively, he could operate the model in the high-resolution mode characterized by the relationships in the area encompassed by the dashed line. In that case, he would have to specify detailed inputs such as range-payload relationships for the generic ground-support aircraft, the theater geometry, delivery tactics, and weapon kill probabilities. Similarly, the user could move to a lower level

of resolution by directly specifying the estimated air-support potential available in main thrust sectors.

Interestingly, the GEFRAM model was not designed with the concept of variable resolution as an explicitly recognized objective. Rather, it was conceived for first-order estimates of a defender's operational minimum in a theater relative to the potential of the attacker's forces (Huber, 1990b, pp. 164 ff). However, with its variable-resolution properties it would also provide a flexible tool for decision support in war games, providing participants with information about critical force ratios as the game unfolds.

In summary, then, internally hierarchical variable-resolution modeling is often straightforward for portions of a system, but it is often difficult to accomplish consistently and comprehensively, especially in complex systems such as the air-land battlefield. Approximations and even exceptions are necessary. Further, variable-resolution zooming cannot be accomplished continuously and arbitrarily; instead, it is a matter of changing from one level to another among those permitted. In some cases, "seamless" variable resolution is possible, but not arbitrary variable resolution.

5. CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

CONCLUSIONS

With this background of observations and speculations, which need to be substantially enhanced by research and analysis that includes more extensive comparisons with the experience of other workers in the community, we suggest the following principles and procedures for the development of new higher-level combat models (e.g., those describing theater- and operational-level phenomena).

- Models (or integrated families of models) should have variable-resolution features that include one or more aggregated levels comfortable to analysts doing work at a parametric level or to other users who don't wish to be bothered with a more detailed simulation of the process at issue. A trivial low-resolution level (e.g., a scalar parameter rather than a parameter indexed by side or theater) will often not be adequate.
- It should be recognized that high-resolution models with highly uncertain data values may be *less* accurate and less useful than aggregated models with fair or good input data, coupled with sensitivity analysis performed on aggregated parameters.¹
- On the other hand, high-resolution models are often superior for communicating and exploring logical connections; they may also be crucial in estimating what range of aggregated parameter values is reasonable, and in some cases the high-resolution parameters have a more intuitive significance than those of lower resolution.
- The *essence* of the variable-resolution design is needed as early as possible in the development process, because attempting to add variable-resolution features after the fact is laden with difficulties. Fortunately, what we refer to here as "the essence" is a top-level structuring, not algorithmic details. Initial design can and should leave multiple stubs to be filled in later.
- Where possible, variable resolution achieved by integrated hierarchical design is preferred because it can lead to a more nearly seamless result.
- New software tools are needed to assist in variable-resolution design. These should include advanced high-level languages and graphical user interfaces that can be used to specify and document major aspects of the code. In the absence of such tools, there will continue to be problems such as side effects, incomplete specification, and confusion.

¹This may be a bitter pill for some modelers/programmers to accept, because they may have done a fine job building a model which is then relegated to the "experimental" category because the data base hasn't yet been scrubbed. However, a beautiful wrong model is still wrong, while a crude parametric model can at least be explicit, controllable, and understandable.

- Specification and documentation need to be top-down and hierarchical; they also need to be in a form communicating efficiently the interrelationships. In practice, this typically means control flow, data flow, state-transition, and influence diagrams.
- The common perception of programmers that there are large differences between models and implementation should be combatted. When such differences exist it is because the programmers have in fact defined the model, which is not acceptable practice unless the model design is adjusted accordingly.

SUGGESTIONS FOR FURTHER RESEARCH

Late in 1991, one of us (Davis) and RAND colleague Richard Hillestad held a DARPA-sponsored workshop to discuss variable-resolution issues with others in the community and to identify critical issues. The participants came from government agencies, federally funded research and development centers, and academia. The principal result of the workshop was a consensus that a full-fledged scientific conference was desirable and that the following issues were good candidates to be taken up in that conference:

- Defining and discussing types and degrees of model "consistency"
- Better understanding the significance of object-oriented modeling and programming for variable-resolution design. What problems do they and do they not solve or mitigate? (We have included some observations on that matter in the current Note.)
- Identifying where variable-resolution methods are most needed
- "Challenge problems," by which was meant problems difficult enough to illustrate the concepts and issues, but simple enough to be worked by different groups and comparisons made across groups
- How to use expert analysts from all fields
- How to do calibrations and adaptations (e.g., via response-surface methods or other sensitivity-analysis methods)
- Nonmonotonic behavior and the related issue of chaos
- The "configuration problem" (Many aggregations produce seriously erroneous results because they destroy information about important spatial relationships among system entities.) (See, e.g., Horrigan, 1991.)
- Mesh size (One aspect of variable resolution involves, for example, the size and shape of spatial grids; these can interact in subtle ways with other aspects of the model.)

- Understanding the "voyage of discovery" (i.e., what is sometimes called "exploratory modeling and analysis," in which one uses models more to understand the relationships and the space of possibilities than to make predictions) (See, for example, Bankes, 1992.)

To this list we would add an emphasis on the design of advanced modeling and analysis environments to improve design, VV&A, and documentation, and to simplify the inevitable adaptations one makes to models that are being seriously used.

RAND will be arranging a DARPA-sponsored conference to address these issues in the late spring of 1992. An objective of that conference will be to create an interdisciplinary setting allowing workers from different disciplines to share experiences and techniques, since variable-resolution modeling is indeed a generic issue of interest in a wide variety of areas.

Appendix A

SYSTEMS ANALYSIS AND NONMONOTONIC BEHAVIOR

A special problem in variable-resolution modeling is that many users of low-resolution models need models with smooth monotonic behavior in some dimensions. For example, when doing cost-effectiveness work, one wants models such that outcome always improves (or at least stays constant) as one adds capability to one's own forces. By contrast, in both simulations and the real world, behavior may *not* be monotonic. To make things worse, the dependence of effectiveness vs. capability may be highly sensitive to factors about which one is fundamentally uncertain (e.g., enemy strategy and tactics, or even the prowess of one's own commanders at various levels).¹

On the one hand, the systems analyst *should* want to know when adding additional capability might plausibly worsen effectiveness (nonmonotonic behavior). Typically, this can arise because of flawed force employment (i.e., flawed strategy and tactics), physical constraints such as crowding when too many aircraft are operating from a given airbase, or political side effects (e.g., loss of allied support because of an operational strategy that causes too many casualties). On the other hand, should defense planning be based on assumptions such as that our generals will be poor performers or that alliance politics will be exceptionally fragile? How much should the nation spend hedging against poor generalship or unwise alliance policies? There is no simple answer except "do the analysis both ways."

One long-standing cultural schism between systems analysts and operationally oriented strategists (military and civilian) has been the tendency for the former to assume effective use of the resources provided, by both enemy and friendly forces, and for the latter to assume a variety of real-world political and doctrinal constraints. Ideally, systems

¹One example here is classic. NATO's defense strategy in the Central Region was, during the Cold War, a relatively rigid forward defense to be implemented with forces that were maldeployed (too few forces in the north and too many forces in the south, where the terrain was already most favorable to the defense). When studies were conducted to see the benefits of adding additional capabilities to NATO forces or, more particularly, to U.S. forces, it was not unusual to find that simulated war outcomes *worsened* with the addition of capability. The reasons were interesting. In some cases, the basis for the result was the model assumption that NATO would conduct the rigid forward defense until it failed, after which it would try to regroup at a river line to the west. However, in scenarios in which the rigid forward defense was doomed, having modest extra capabilities propped up the fatally flawed strategy long enough for more reserves to be sent to the front where they would be lost when breakthroughs occurred. With enough extra capability, of course, the defense would succeed, but overall behavior was nonmonotonic. In other cases, the basis for the counterintuitive result was that if one improved U.S. defenses, the Pact forces would focus more exclusively on non-U.S. sectors, which was wiser in any case because of their relative weakness. In all of these cases, then, the counterintuitive results stemmed from flawed force employment. Unfortunately, the flaws may have been realistic.

analysis should employ a mix of "realistic," nonmonotonic models and somewhat idealized models showing the potential value of various improvement options. The idealized models would exhibit monotonic behavior and be useful in cost-effectiveness comparisons, but results would be presented with appropriate cautions and auxiliary recommendations based on the more "realistic" work,² including the use of *multiscenario analytic war gaming* in the broadest sense (Davis, 1986, 1988).

The relationship of all this to the current Note is that variable-resolution modeling as we describe it is motivated in part by the need to provide options for analysis. However, analysts often need well-behaved (monotonic) effectiveness vs. capability relationships, which often require aggregated models and related decision models that employ resources well.³ Even such models may not be entirely monotonic without further smoothing, because of a variety of decision-model thresholds, discrete time steps, terrain-zone boundary effects, and so on.⁴ In addition, analysts may need models that are more realistic. Thus, in developing variable-resolution designs, we may have to consider having alternative low-resolution submodels within a given hierarchical process. These might differ, for example, only with respect to allocation-of-resource subroutines. Or they might differ in many other respects as well.

To illustrate the importance of having monotonic behavior, consider Figs. A.1 and A.2, both of which display effectiveness vs. expenditure for each of two options A and B (e.g., pursuing options A and B might correspond to buying more heavy and light divisions, respectively). Interpreting Fig. A.1 is easy: Option A appears to be superior. By contrast, Fig. A.2 is not. To make things worse, it is often the case that one conducts simulations at only a few points because of long run times or the demand for quick answers. As the points in Fig. A.2 illustrate, with this type of nonmonotonic behavior, one could readily conclude, incorrectly, that Option B was superior.

²For defense of this approach see Hawkins (1984), Davis (1985), and Davis (1988).

³Examples of decisionmaking modules include: (a) the use of linear programming (e.g., the Arsenal Exchange Model) to develop optimal weapon allocation strategies for both Red and Blue in classic strategic-nuclear analysis; (b) the use of more complex methods to optimize the allocation and apportionment of tactical air forces in theater models (e.g., the SAGE algorithm developed by colleague Richard Hillestad); and (c) the use of knowledge-based decision models to assure reasonable, albeit nonoptimal, adaptation of air and ground force employment in theater studies (e.g., the decision models of the RAND Strategy Assessment System described in Davis and Howe, 1990).

⁴For relevant discussions see Hawkins (1984) and Farrell (1984), who use the term *structural variance* to mean nonmonotonic variance of results due to structural features of the underlying model, and Dewar, Gillogly, and Juncosa (1991), which discusses nonmonotonic behavior and even chaotic effects in simple combat models. Nonmonotonicity effects have been found in a variety of well-known models including Vector, Vic, and the RSAS. They are almost certainly common, even if not commonly discussed.

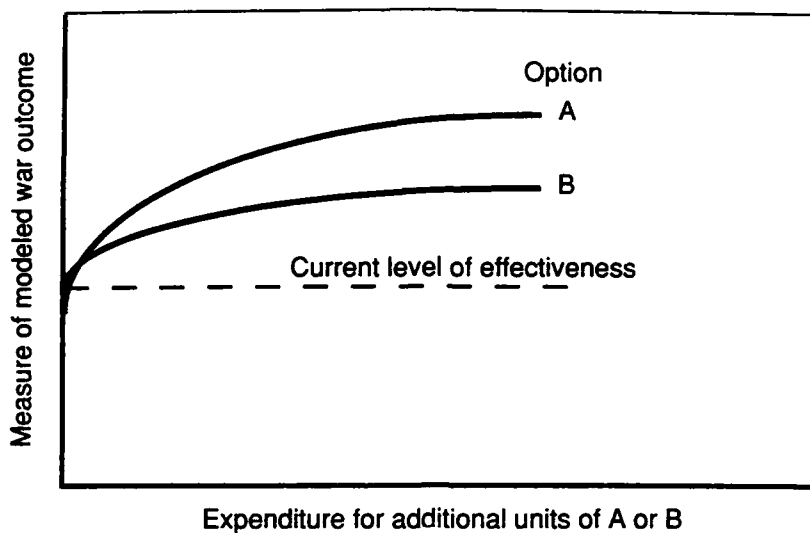


Figure A.1—Monotonic Behavior of Outcome vs. Capability for Options A and B

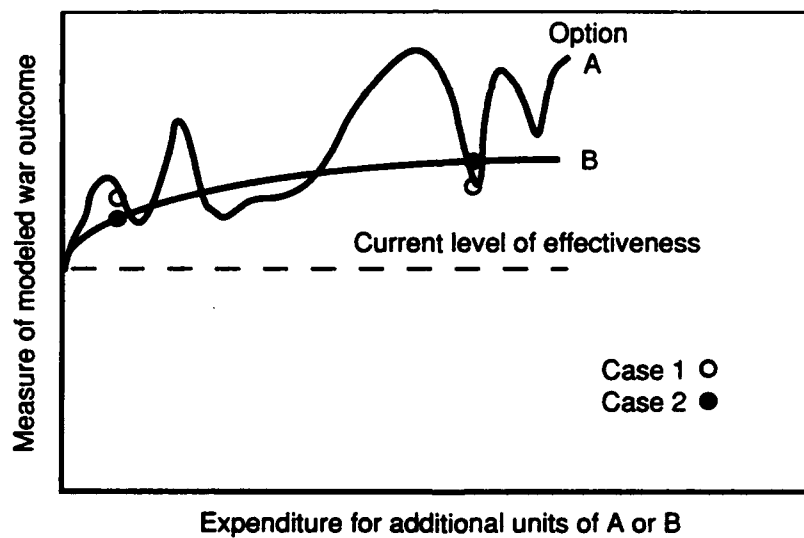


Figure A.2—Inconsistency in Systems Analysis Due to Nonmonotonic Model Behavior

Appendix B

HISTORICAL BACKGROUND

The general notion of variable-resolution combat modeling has existed since at least the early 1980s. Its origins are related to a modeling concept commonly called the *hierarchy of models* pursued since the mid-1970s by several analysis institutions in Europe (notably the German IABG) and the United States. Importantly, these hierarchies were not integrated in the sense we discussed above. Instead, the term hierarchy applied only because there was a high-level model and some lower-level models, with the latter being used to inform the former.

The basic motivation in this early work was to keep model complexity at manageable levels as the scope of analysis extended from weapon-on-weapon issues through tactical-level combat, operational-level combat, and finally strategic-level conflict. The idea was that combat simulation models developed independently to deal with the separate levels could be linked so that a model at a given level would provide inputs for the simulations at the next higher level (uplink) and scenario parameters for the model at the next lower level (downlink).

To use terminology of the early 1980s,¹ the linkage between models in a hierarchy may be either "external" or "internal." External linkage is provided through data generated outside the hierarchy (i.e., through off-line analysis) by means of aggregation or disaggregation techniques, or other models. Internal linkage requires that the respective models are truly *integrated* into the hierarchy—i.e., the models fit together in terms of the substantive content. Only an internally linked model hierarchy should be properly classified as a variable-resolution model with the capability of aggregation and disaggregation, but usage varies.² In the text we use "internally linked hierarchy" and "integrated hierarchy" to mean the same thing.

¹This distinction was drawn in a 1982 NATO conference surveying the state of the art in combat modeling (Huber, 1984). The model hierarchies presented at the conference, by the U.K.'s DOAE and Germany's IABG were externally linked. Internal linkage was discussed for the model hierarchy concept proposed by the U.S. Army's Army Model Improvement Program (AMIP). See Robinson and Fallin (1984).

²Parry (1984) speculated about such an internally linked model hierarchy, primarily as a response to what he and others in the 1982 NATO conference saw as serious problems in the U.S. Army's approach to the hierarchy of models concept, one that sought to employ (and to some extent has employed) independently developed models.

It appears not to have been recognized, or at least discussed, in the early 1980s, that the capability to step up or down in resolution would also result if the models were designed in a quasi rigorously hierarchical manner permitting one to achieve different resolutions by truncating or expanding the model tree at the desired level³—what we referred to in the text as integrated hierarchical design of a single model. In this case, the variable-resolution capability (IHVR) is an inherent feature of the model design. That is, by virtue of the strictly integrated hierarchical design, a variable-resolution capability follows straightforwardly. Although the idea may well have been developed in numerous other places over the years, it was developed independently by one of us (Davis) as part of mid-1980s research on the RSAS. Portions of the RSAS have IHVR-style variable resolution, as discussed in the text; other portions do not. We believe the concept of IHVR in a single model is very useful even if one then chooses to have an integrated hierarchy of separate models, because the principal design issues of achieving substantive integration are the same as for achieving this integration within a single model.

In metaphorical reference to cameras, variable-resolution models are frequently considered to have zooming capability, glossing over the fact that resolution in models can typically be changed only in steps, not continuously. Table B.1 presents a taxonomy of zooming methods using the terminology of the early-1980s' discussions of variable resolution. In this taxonomy, only IHVR and selected viewing are considered to imply a zooming capability, and in the latter case it is unidirectional only, in the form of an interface between a high-resolution model and a human analyst or viewer who wants an aggregated picture of the evolving situation.

The change of resolution by selecting an alternative submodel does not fit this taxonomy. With reference to cameras, it corresponds more to a change of lenses and filters than to zooming. It may indeed be questioned whether selecting alternative submodels should properly be thought of as a class of variable resolution, because one can argue that it is merely a reconfiguration of a model resulting in a different model.

Except for some submodels discussed in the text, none of the presently operational combat models of which we are aware have a zooming capability, because all are externally linked (i.e., they are connected only through off-line analysis and data manipulations). This is true for the ground-combat models of the U.S. Army's Concepts Analysis Agency (CAA), Germany's IABG, a number of U.S. air-force models used in Air Force Studies and Analysis and in RAND studies, and the hierarchy of ground-combat models currently used at RAND

³We use the term quasi rigorous here because in practice many and perhaps most complex military models cannot be made completely and rigorously hierarchical. This is discussed in the text.

Table B.1
Types of Zoom Capability

Mechanism for Zooming	Direction	Design of Model	Purpose of Zooming
Inherent feature of model	Zooming in	Internally hierarchical (inherently integrated)	Higher-resolution simulation
	Zooming out		Lower-resolution simulation (selective viewing)
Interface models	Zooming in	Hierarchical family of integrated models	Disaggregation of simulation results (downlink between models)
	Zooming out		Aggregation of simulation results (uplink between models)
		High-resolution models	Aggregation of simulation results (selective viewing)

(Gritton, Don, and Steeb, 1990). Whether these models could be linked internally by the integration of newly designed interface models is very much open to question; it would probably be quite difficult in most cases, because having a hierarchy of models does not ordinarily mean that the models have been purposely designed to operate together.

Appendix C
AN EXAMPLE OF CONSISTENCY IN AGGREGATION

Simply to illustrate certain aggregation issues, consider the extremely simple problem of N falling bodies, each of which is described by Newton's law with a deceleration term representing drag, assumed to be proportional to speed. For each body, we then have:

$$\frac{d}{dt} h_i(t) = -v_i(t) \quad (1)$$

$$\frac{d}{dt} v_i(t) = g - D_i v_i / m_i \quad (2)$$

The reader now should ask himself whether he would expect a similar set of equations to apply for the altitude and velocity of a hypothetical "average" body. That is, is there a good aggregation of this problem? If there is, what form do the equations take? Our experience is that people generally have poor intuition on this matter, even though the problem is quite simple. Some people expect the aggregation to work; others do not. Let us now solve the problem and discuss it.

Although the problem has a closed-form analytic solution so long as the drag coefficients and masses are constant, let us approach the problem as we would in a simulation program, by applying the following equations iteratively with an appropriately small time step. Until the body hits the ground, we have

$$\begin{aligned} h_i(t + \Delta t) &= h_i(t) - v_i(t)\Delta t \\ v_i(t + \Delta t) &= v_i - g\Delta t + (D_i v_i(t) / m_i)\Delta t \end{aligned} \quad (3)-(4)$$

It is rigorously true, at all times, that the average altitude and average speed of the falling bodies are given by:

$$\begin{aligned} \bar{h} &= \sum_i^N h_i / N \\ \bar{v} &= \sum_i^N v_i / N \end{aligned} \quad (5)-(6)$$

It follows, then, that the exact equations for the aggregated quantities (i.e., the altitude and velocity of the "average" body) are

$$\begin{aligned}\bar{h}(t + \Delta t) &= \bar{h}(t) + \bar{v}(t)\Delta t \\ \bar{v}(t + dt) &= \bar{v}(t) - g\Delta t + \bar{H}(t)\Delta t \\ \bar{H}(t) &= (1/N) \sum_i^N D_i v_i(t) / M_i\end{aligned}\tag{7)-(9}$$

Comparing Eqs. (7)-(9) with Eqs. (3)-(4), it is clear that in the general case the aggregation does not follow the same equations as the equations for the individual bodies. However, let us now consider approximations. A standard approximation would be to "break the average" in Eq. (9) by assuming that the "average of the product is the product of the averages." We would then have:

$$\bar{H}(t) \cong \bar{D}(1/N) \sum_i^N v_i(t) / M_i\tag{10}$$

And, taking this a step further by replacing M_i with the average mass, we have

$$\bar{H}(t) \cong \bar{D} \bar{V}(t) / \bar{M}\tag{11}$$

Whether this is a good or poor approximation is not obvious a priori, since it depends on the nature of the individual bodies. The point of this example is that the form of Eqs. (3)-(4) is identical to the form of Eqs. (7)-(8) if and only if one makes these approximations. If they are good approximations, then we say the aggregation works. If they are not, it does not. There is nothing very deep about the matter, nor anything for people to argue about (i.e., no reason to argue about whether aggregation as a philosophy is good or bad). That is, one might be suspicious of aggregation, but the aggregation might nonetheless be a very good approximation indeed.

To illustrate this, Fig. C.1 shows the time histories of three falling bodies. In this instance, the drag coefficients were assumed to be in the ratio of 1:6:2 and the masses were assumed to be equal. Figure C.2 compares the exact time history of average altitude with the time history computed using the approximation in Eq. (11).

As we see, the agreement is very good indeed for the first 11 seconds. It is then rather bad in the last several seconds. The real point here is that *one's intuition is often not adequate to judge whether behavior in the aggregate will have a simple behavior, or even a*

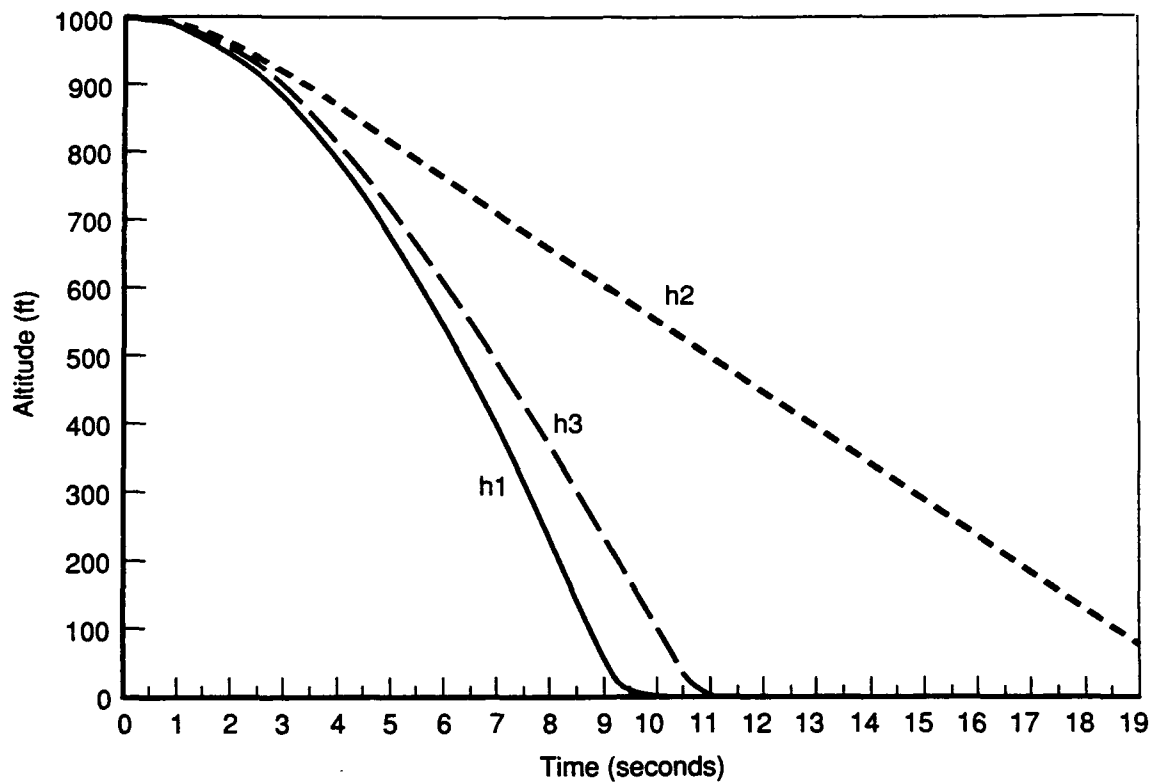


Figure C.1—Altitude vs. Time for Separate Falling Bodies

behavior mathematically similar to behavior of a component. Typically, the degree of consistency depends on numerical details. There are domains of consistency and domains of inconsistency.¹ Our second point, to be expanded upon in subsequent work, is that there is little tradition in combat modeling of developing explicit approximations that include estimates of the error and its implication in the context of the policy-relevant calculation. As a result, arguments about aggregation are often somewhat mindless.

¹Note that a different approach to low-resolution modeling would be to fit the exact results to a regression equation. Agreement would then be better, but the physical significance of the functional form of the resulting low-resolution "model" would then be obscure. In this case, an improved model could be developed by doing a series expansion around the approximate solution, and thereby generating approximate corrections to Eq. (11). Or, as mentioned above, one could use a closed-form solution in the special case of constant mass and drag coefficient.

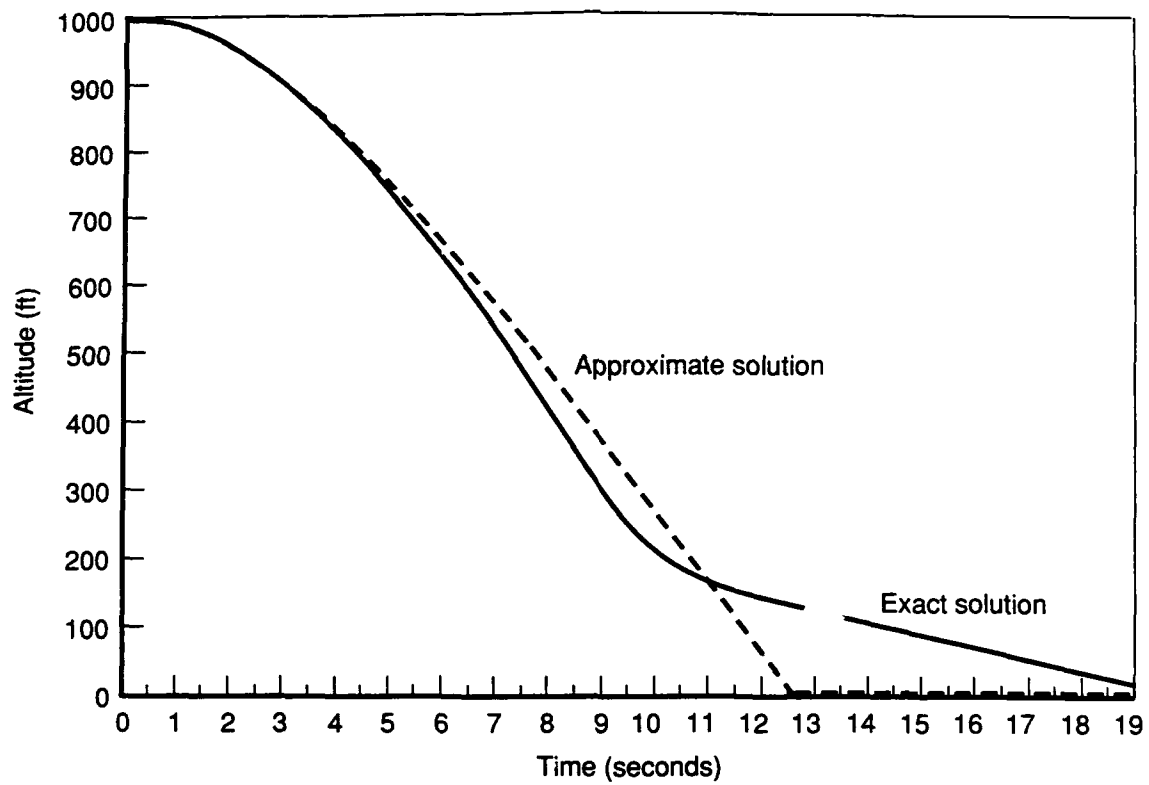


Figure C.2—Aggregated vs. Exact Model Predictions of Average Altitude vs. Time

BIBLIOGRAPHY

- Allen, Patrick (forthcoming), *Evolution of Models at the Warrior Preparation Center*, RAND.
- Bankes, Steven C. (1992), *Exploratory Modeling and the Use of Simulation for Policy Analysis*, RAND, N-3093-A.
- Bennett, Bruce W., Carl M. Jones, Arthur M. Bullock, and Paul K. Davis (1988), *Main Theater Warfare Modeling in the RAND Strategy Assessment System (3.0)*, RAND, N-2743-NA.
- Cherry, W. Peter (1984), "Vector-2 Land/Air Campaign Model," in Huber (1984).
- Davis, Paul K (1986), "Game-Structured Analysis as a Framework for Defense Planning," in Huber (1986), based on a conference in 1984. Published also as RAND, N-2751-RC, 1988.
- Davis, Paul K. (1987), "A New Analytic Technique for the Study of Deterrence, Escalation Control, and War Termination," in Stephen J. Cimbala (ed.), *Artificial Intelligence and National Security*, Lexington Books, D.C. Heath and Co., Boston, Massachusetts.
- Davis, Paul K. (1988a), *The Role of Uncertainty in Assessing the NATO-Pact Central-Region Balance*, RAND, N-2839-RC.
- Davis, Paul K. (1988b), *Explanation Mechanisms for Knowledge-Based Models in the RAND Strategy Assessment System*, RAND, N-2711-NA.
- Davis, Paul K. (1989), *Modeling of Soft Factors in the RAND Strategy Assessment System (RSAS)*, RAND, P-7538.
- Davis, Paul K. (1990), *An Analyst's Primer for the RAND-ABEL Programming Language*, RAND, N-3042-NA.
- Davis, Paul K., and Donald Blumenthal (1991), *The Base-of-Sand Problem: A White Paper on the State of Military Combat Modeling*, RAND, N-3148-OSD/DARPA.
- Davis, Paul K., and H. Edward Hall (1988), *Overview of System Software in the RAND Strategy Assessment System*, RAND, N-2755-NA.
- Davis, Paul K., and Robert D. Howe (1990), *Representing Operational Strategies and Command-Control in the RAND Strategy Assessment System (RSAS)*, RAND, N-3232-OSD.
- Davis, Paul K., Steven C. Bankes, and James P. Kahan (1986), *A New Methodology for Modeling National Command Level Decisionmaking in War Games and Simulations*, RAND, R-3290-NA.
- Dewar, James A., James J. Gillogly, and Mario L. Juncosa (1991), *Non-Monotonicity, Chaos, and Combat Models*, RAND, R-3995-RC.

- Farrell, Robert L. (1984), "How Non-Weapon-System Parameters Affect Results," in Huber (1984).
- Fisher, Gene H. (1970), *Cost Considerations in Systems Analysis*, RAND, R-490-ASD.
- Goeller, Bruce, et al. (1983), *Policy Analysis of Water Management for the Netherlands*, RAND, R-2500/1-NETH.
- Gritton, Eugene C., Bruce W. Don, and Randall Steeb (1990), *RAND's Concept Analysis Environment: A Briefing*, RAND, N-3004-DARPA.
- Hawkins, Geoff (1984), "Structural Variance and Other Related Topics Experienced in the SHAPE Armour/Anti-Armour Study," pp. 627-635, in Huber (1984).
- Horrigan, Timothy (1991), *Configural Theory and the Mathematical Modeling of Combat: Realizing the Potential of Models and Simulations of Combat to Improve the Effectiveness of Weapons, Tactics, and Training*, Horrigan Analytics, HAS 91-17-1.
- Hosmer, Bradley C. (1988), "Operational Art: The Importance of the Operational Level of War," *Phalanx*, Vol. 21, No. 3.
- Huber, Reiner K. (1985), "On Current Issues in Defence Systems Analysis and Combat Modelling," *OMEGA*, Vol. 13, No. 2, pp. 95-106. This appeared earlier in Huber (1984), pp. 11-34.
- Huber, Reiner K. (editor) (1984), *Systems Analysis and Modeling in Defense: Development, Trends, and Issues*, Plenum Press, New York.
- Huber, Reiner K. (1990a), "Systemanalytische Instrumente der Sicherheitspolitischen Analyse und Strategie Planning," pp. 481-503 in *Strategie-Handbuch*, Band 2.
- Huber, Reiner K. (editor) (1990b), *Military Stability—Prerequisites and Analysis Requirements for Conventional Stability in Europe*, NOMOS Verlagsgesellschaft, Baden-Baden, Germany.
- Hughes, Wayne P. (ed.) (1989), *Military Modeling*, Military Operations Research Society, 2d edition, Alexandria, Virginia.
- Parry, Sam H. (1984), "A Self-Contained Hierarchical Simulation Construct," in Huber (1984), pp. 547-558.
- Quade, E. S., and W. I. Boucher (1968) (editors), *Systems Analysis and Policy Planning: Applications in Defense*, RAND, R-439-PR.
- Rehms, Paul (1991), "The Warrior Preparation Center," *PHALANX*, Vol. 24, No. 2, pp. 13-15.
- Robinson, John D., and Herbert K. Fallin (1984), "U.S. Army Model Improvement Program (AMIP)," in Huber (1984).
- Rothenberg, Jeff (1990), *Prototyping as Modeling: What Is Being Modeled?* RAND, N-3191-DARPA.

- Rumbaugh, James, Michael Blaha, William Premerplani, Frederick Eddy, and William Lorensen (1990), *Object-Oriented Modeling and Design*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Schwabe, William, and Barry A. Wilson (1990), *Analytic War Plans: Adaptive Force-Employment Logic in the RAND Strategy Assessment System (RSAS)* RAND, N-3051-NA.
- Shapiro, Norman Z., H. Edward Hall, Robert H. Anderson, Mark L. LaCasse, Marrietta S. Gillogly, and Robert E. Weissler (1988), *The RAND-ABEL Programming Language: Reference Manual*, RAND, N-2367-1-NA.
- Simon, Herbert (1982), *Models of Bounded Rationality*, MIT Press, Cambridge, Massachusetts.
- Weatherly, Richard, David Seidel, and Jon Weissman (1991), "Aggregate Level Simulation Protocol," *Proceedings of the Summer Computer Simulation Conference*.
- Zeigler, Bernard (1990), *Object-Oriented Simulation with Hierarchical, Modular Models*, Academic Press, New York.