Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

THE SCIENTIFIC METHOD OF CHOOSING MODEL FIDELITY

Michael P. Bailey
William G. Kemple

Department of Operations Research
Naval Postgraduate School
Monterey, CA 93943, U.S.A.

ABSTRACT

Simulation modeling currently enjoys great popularity as a tool for solving problems within Department of Defense activities. In this work we consider the process of upgrading an existing simulation model by increasing the fidelity of the model. The submodels to be upgraded and the degree to which they are upgraded should be chosen in a coherent, scientific manner. This is currently not the norm.

In this work we describe a method which a simulation analyst can use to choose from a set of proposed model upgrades that accounts for both the costs of the upgrade as well as the benefits.

1 INTRODUCTION

Emerging computer environment standards and technology, as well as software design improvements, have produced an unprecidented opportunity to improve existing models used in support of military analysis. The model enhancements selected for implementation come from a set of initiatives generated by the developer/user community. Although the designers, developers, and users of simulation models are sophisticated technologists and decision scientists, the model improvements they select are usually chosen in an entirely unscientific way.

In this work, we present a fledgling methodology proposed for determining which aspects of a given model should be chosen for improvement, and discuss the implementation of this method using examples from experiences in model development.

Emerging technologies in the computer science realm have produced an opportunity to change model fidelity expediently. Simulation modeling software using the object-oriented programming paradigm has existed for decades, see Bertwistle (1973), but object-oriented models have a reputation for being poor performers. A diversity of products have now become available which combine modern software production environments, object-oriented simulation (OOS), and performance competitive with FORTRAN models.

The relevance of this OOS revolution to the present discussion is that object-oriented models are forced to be modular, so that exchanging an existing submodel with an enhanced one does not carry the cost it used to. Thus, growth of simulation models from primative to sophisticated is more graceful, costs less, and does not rely on personnel possessing extreme intimacy with the model's implementation.

Computer development and runtime environments are finally becoming standardized, an initiative commonly known as *open architecture*. The impact of this movement is that, because of the degree of standardization, models are portable to an unprecidented degree. Models must no longer adhere to the political boundries of the computer environment map. Model developers are now capable of producing machine-independent implementations so that models can keep pace with the growth of computer environment capabilities. Improving model performance and capabilities no longer comes at the high cost of reimplementation.

In this environment, model extensions can be pursued by organizations other than the model's sponsor. Hence, the sponsor will be presented with a variety of model upgrades whose software development costs are **sunk costs**. The sponsor can select a subset of the already-implemented upgrades for inclusion in the supported model. In this environment of continual upgrades with negligible software development costs, the model's sponsor organization should make upgrade choices in a structured manner.

The changing nature of the perceived threat and combat environment will motivate organizations to expand models to accomodate scenarios different from those for which the models were designed. Reduced testing budgets place more pressure on the modeling community to produce tools capable of reliably evaluating weapon systems' performance, tactics, decisions, and policies.

Thus, progressive upgrades of existing models are both feasible and necessary. *Which submodels* to improve is a choice we will face more often, and one we should deal with scientifically.

2 FIDELITY

In this section, we attach some formality to the notion of model fidelity. Some underlying assumptions concerning the nature of the level of fidelity problem will be presented so that the subsequent mathematical formulation is understood.

2.1 Definitions

Model **level of fidelity** refers to the degree to which the model produces the same outcomes as the tangible, physical system. Thus, a policy constructed with the aid of a model with infinite fidelity would be identical to a policy produced using unlimited experimentation with the real system.

Model **validation** (as we define it) is the practice of comparing the model to the physical system and concluding that the model produces outcomes similar to the physical system. The hope is that the model output is similar to the physical system to the degree that the model is usable. Model validation is often confounded by several factors:

- the physical system is not observable because it does not yet exist, observing it is a hazard to the system or the observer, the agency in control of the system prevents it from being observed, or observation itself causes the system to change its behavior;

- the environment for which the physical system is intended does not yet exist, or is inaccessible for some reason, or the physical system does not operate in an environment as pristine or consistent as that generated by the computer;

- the model is steady-state, a condition not attained by most physical systems.

Thus, model validation is often impossible in the military modeling discipline.

If we loosen the definition of validation to include the practice of *conceptual* or *subjective* validation, see Balci (1990), we can validate more models but only at the expense of objectivity and user trust – the reasons we pursue validation. The usual response to this situation is to increase **model resolution**. Here we define resolution to be degree to which detail is included in submodels. In the familiar barbershop queuing model seen in most introductory simulation courses, the system is modeled as a M/M/n queue. If we introduce

- different customer types (styles, crewcuts, fades);

- different barber speeds;

- time-of-day effects;

- opening and closing the shop;

- earthquakes;

we are adding resolution to the model. In object-oriented models, increasing resolution means replacing larger objects by semiautonomous subobjects, replacing simple decision logic with more complex logic, using more source data such as higher resolution terrain data, including more objects, or simply improving approximations. Increasing resolution makes submodels more data-dependent.

Summarizing, validation of military models is often weak or impossible, models of high fidelity are valid, models of high resolution may or may not be valid but are always complex. Resolution is often (poorly) used as a surrogate for fidelity.

2.2 Assumptions

The hidden assumption in the approach we take is that model fidelity *must not* decrease with model resolution. Simulation modeling may be thought of as the modeling of the interactions of objects and their environment. As the resolution becomes greater, the logic within an object becomes more environmentally dependent because more of the system is considered environmental to the (smaller) object. Thus, user confidence grows with resolution because more of the physical system is taken into account by the actions of the model's objects.

Is this reasonable? Analysts have often found themselves arguing in the negative, often to the frustration of both themselves and the sponsoring organization. The sponsor cannot fathom why the analyst insists on ignoring physical realities of the system modeled, while the analyst sees resolution as a source of obfuscation. The method proposed by the analyst is often much too mathematically sophisticated to be useful for the sponsor. Intercession in this often-occuring debate is critical to the future growth of the modeling community.

2.3 Example

In Bailey *et al* (1992), an object-oriented simulation model of a Marine Air Ground Task Force (MAGTF) command, control, communications, computers, and intelligence (C^4I) network, called the Marine Corps Communications Architecture Analysis Model (MCCAAM), is presented. The Marine Corps is preparing to procure a next-generation single-channel radio with anti-jamming capability. MCCAAM was designed to evaluate performance of allocations of next-generation radios to units in the MAGTF, where the measure of performance of an allocation is tactically driven. The ultimate goal is to select the best mix and allocation of radios.

In this work, we will describe a set of three possible upgrades to the MCCAAM model which are already implemented. They can be looked on as switches which the user can turn to increase or decrease the resolution of MCCAAM in three areas, they are

1. the presence of radio jammers;

2. the failure and repair of radios;

3. the persistence of communication tasks.

The presence of jammers would seem to be a step in the right direction, as the new radio being deployed has anti-jamming capability while the existing radio does not. The failure and repair of radios might be critical to the fidelity of the model, as the new radios have better reliability performance, but take longer to repair and take more time to reenter service.

When a communication task is not completed in time, the system assesses a penalty per unit time late for the task. If the tasks are persistent, they assess this penalty until they are completed. However, if we determine that the task has been overcome by events, we can stop the task from continuing. This action stops the assessment of penalty, as well as decongesting the communications network.

In what follows, we will assess the costs and benifits of including these three upgrades in MCCAAM, and describe how we can come to a conclusion about the efficacy of making the upgrades.

3 ASSESSING COSTS AND BENEFITS OF INCREASING FIDELITY

Why aren't current models of extremely high resolution? The answer is fairly obvious, though sometimes difficult to elucidate. Model developers have cultivated a framework of constraints regarding the capabilities of simulation, an intuition concerning the level of importance of a submodel to the performance of the overall system, the availability of data, and the sophistication of the user. Finally, the budget afforded a given development effort dictates certain constraints on the fidelity of a model. This framework must be challenged, as it represents inertia which the sponsors cannot afford.

3.1 The Impact to Usability of the Model

Any constraint in model fidelity reflects some cost. As stressed above, development costs will not be major contributors to the negative impacts of model upgrade. Thus, what are these negative impacts, and what do we include as a negative impact?

3.1.1 Data Hunger and Risk

It is clear that one of the major shortcomings of an increase in resolution is an attendant increase in data hunger. As a submodel becomes more specific and detailed, the requirement for supporting data increases. Collection of the appropriate data to support a submodel is vulnerable to the same shortcomings as model validation. Furthermore, some of the required data is often used to support some judgemental process, such as the probability that a missle battery acts on a partially jammed tracking solution.

An axiom of model development is that more resolution requires more data. Sponsors do a great deal of data collection for the systems they administer, but this data is not usually collected with modeling in mind. Architecture standards should force all of these data sources to be compatable, so that access to data will not represent much cost. Thus, the **risk** in data sources is the major cost of using data.

3.1.2 Performance Degradation

Closely associated with the resolution of the supporting data used, poorer performance of the model is seen as a negative impact of model fidelity increase. While it is certainly true that computers are getting faster, increasing resolution of a submodel can cost several orders of magnitude in algorithm complexity. The analyst who develops a model's upgrades must evaluate the impact each upgrade has on the computational budget.

3.1.3 Increased Developer/User Sophistication Required

By including more detail in a submodel, the developer increases the required level of understanding for himself and the user. To a great extent, the user can approach submodels he does not understand as *black* *boxes*, and experiment only with the processes he understands. This approach carries some risk of model misuse, but defensive software design can reduce this. Some consideration must also be given to the overall comprehensibility of the model.

The model developer's level of system knowledge is not optional. In order to produce the appropriate submodel, the developer must be expert in the model and in the system. Aggregated models are often produced because the modeler does not have the background, the appetite, or the time to become a system expert. This problem is particularly prevalent in models produced in an academic environment, where depth of knowledge of the physical system may not be a high priority.

3.1.4 Measuring Cost

Measuring uncertainty of the data sources for alternative submodels is fairly straightforward. There are well established methods for determining the sensitivity of a model to a single datum, see Kleijnen (1975), Cogliano and Schruben (1987), or Glasserman(1990) for academic treatments of this topic. In Rothenberg (1989), a simpler though less developed method for estimating model sensitivity to submodels is presented. By combining sensitivity and uncertainty measures in a manner appropriate for the problem, relative risk can be computed for each upgrade option. By analyzing these values, program managers can constrain this summed risk, so that any set of upgrades undertaken does not exceed the comfort level of the program manager.

Suppose that we are considering a package of submodel upgrades involving submodels $1, 2, \ldots, N$. If the risk associated with submodel i is given as r_i, then the combined risk is $r_N = \sum_{i=1}^{N} r_i$. The project director's comfort level, given by R, constrains the data-hunger-generated risk as $R \geq r_N$ We must remind ourselves that this risk is to the integrity and reliability of the model, not to the developing organization.

Accessing performance costs is straightforward. We want the model to produce the desired outcomes in a specified amount of time. Upgrade negative impact in terms of time should be seen as a multiplier. Thus, if we have upgrades $1, 2, \ldots N$, and testing upgrade i seems to make the time until completion m_i longer, then the performance cost of including the N submodels is $m_N = \prod_{i=1}^{N} m_i$. If we expect our new platform to provide 1.5 times the speed of the current platform, and we cannot sacrifice any turn-around time for the model, then we must have $m_N \leq 1.5$.

Finally, in considering cost of sophistication, measurement seems very difficult. Certainly, budgetary and personnel constraints arise from the developer sophistication cost. User sophistication, to the extent that it is a problem, must be addressed. Certainly, making a simulation model *sailor-proof*requires extra design and implementation assets.

3.1.5 Costing the MCCAAM Upgrades

MCCAAM's three upgrades each need to be evaluated for cost in each of the three categories, data risk, performance, and user sophistication.

We start with the data risk costs. To add jammers to the MCCAAM model, we need the following data:

- location, number and composition of the enemy jamming force;
- technical parameters for each jammer type, including jammer power, bandwidth, dwell time, antenna type, search method, reliability, and duty cycle;
- employment tactics for each jammer.

Additionally, the presence of jammers makes the positions of the receiving units more critical. Based on the size of the baseline MCCAAM database, the size of the database of the upgraded model is only 5% larger. However, if we consider each data item's source, scoring as follows:

current technical pub. or field test	risk = 0
old technical pub.	risk = 1
expert judgement	risk = 5
developer judgement	risk = 10

By scoring the baseline and jammer upgrade models, we find that the data risk score grows by 30%. We made no attempt to determine the sensitivity of the model to the data.

To add radio failure and repair processes to MCCAAM, we performed similar scoring. Because the frequency-hopping radio is new, and the existing radio is a well-tested workhorse, the failure data had no risk. The repair data was drawn from some expert judgements. When a radio does fail, its replacement must reenter the radio net vacated by the broken radio. This reentry process was designed by the developers for the frequency-hoppers. Failure and repair of radios inflated the data risk score by 7%.

To make the perishability of communication tasks, the data required is the allotted time for each kind of task, drawn from our own judgements, and inflating the risk score by 12%.

The performance degradation costs can be accessed objectively. Based on some simple timing experiments, we see that the upgrades slow the baseline model by:

- jammers: 8%;
- failure and repair: 1%;
- task perishability: 4%.

None of these upgrades grossly inflate the turnaround time of the model. The perishability upgrade inflates computation much more than expected because it causes MCCAAM to do a good deal of garbage collection.

The addition of jammers in the model meant that we were required to interview some experts on ground-based jamming operations in order to design the jammer objects. We were also compelled to construct a set of library jammer objects with default data, but they must be taylored to the scenario the user constructs. Thus, we have reduced our potential user population somewhat by making MCCAAM more sophisticated.

The user is not really burdened with any further requirements for expertise for either of the two other upgrades.

3.2 Benefits

From the above definitions, we can say that we seek fidelity, yet we would like it to come at the expense of small increases in submodel resolution. How is the benefit of the increase in fidelity manifested? How do we measure it? We might call any such measurement a measure of overall model fidelity.

In this discussion, we will consider the case where the simulation is used in support of making some selection from a set of alternatives, (eg. C^{4}I architectures, ammunition round reliabilities, EA-6 jamming assignments, or submarine tactics). A different approach must be developed for simulations used for training.

Let $i = 1, 2, \ldots, S$ be the set of feasible alternatives for the selection. Suppose that we *could* experiment with the real system as if it were a simulation model, and that we could sample an unlimited number of replications. Let p_j be the relative frequency of the event that s_j is the best of the S selections. Depending on the underpinning of the reader's philosophical framework of probability and optimization, p_j could be seen as the probability that selection s_j is the optimal choice. Henceforth, treat $p_j : j = 1, 2, \ldots, S$ as probabilities.

We run our selection process for M independent runs for the current model and a model with submodel i upgraded, heretofore known as the baseline and upgraded models, resp.

Let $f_b(s_j)$ be the frequency with which the baseline model chooses selection s_j as the solution, while $f_u(s_j)$ is the frequency of s_j for the upgraded model. $f_b(s_j)/M$ is the baseline's point estimate of p_j, so that the value of the upgrade may be measured by

$$X_u = \sum_{j=1}^{S} \frac{[f_b(s_j) - f_u(s_j)]^2}{f_u(s_j)} \qquad (1)$$

where we use $f_u(s_j)$ as our estimate of p_j. Under the assumption that the upgrade makes no difference in the selection process, we know $X_u \sim \chi^2_{S-1}$, see Agresti (1990). We can construct

$$B_K = \sum_{u \in K \subset 1,2,\ldots,S} X_j. \qquad (2)$$

If we assume (unreasonablly) that the upgrades' effects are mutually independent, B_K is the sum of independent χ^2_{S-1} random variables, so $B_K \sim \chi^2_{|K|(S-1)}$.

3.3 Trading Cost and Benefit

The costs and benefits elucidated above suggest several ways to proceed. Suppose that we

- restrict total data risk score to R;
- restrict performance degradation to D;
- insist that no upgrade demands more knowledge of the developers or users than they can be expected to attain in a short time.

If we let R_b be the data risk involved with the baseline system, and D_b be the run time of the current system, we could proceed as described below

- start with baseline = existing system, R_b = 100%, D_b = 100%, and all upgrades under consideration.
- REPEAT
  1. test run each remaining upgrade to the current baseline system;
  2. find B_K for small sets of upgrades which do not violate the three constraints for data risk, performance, and sophistication;
  3. add upgrade set K with highest B_K;
  4. recompute R_b, D_b for the new baseline;

5. remove all ugrades which are infeasible additions to the new baseline;

- UNTIL no more upgrades are feasible.

3.4 Assessing MCCAAM's Upgrades

As mentioned, the set of possible decision outcomes for MCCAAM analysis are frequency-hopping radio allocations. Our choices are

1. forward edge of battle (FEBA) gets priority in allocationg frequency-hoppers;
2. top command levels get priority;
3. don't use the new radios at all.

Potentially, we can replace 30% of all the radios with frequency hoppers. Suppose that we let constraint right-hand sides be $R = 145\%, D = 115\%$, chosen to make the exposition interesting. The baseline model had

X^2_{JAM}	109.02
X^2_{R+F}	21.10
X^2_{PER}	6.20

As all of the upgrades are feasible additions to the baseline model, so we chose the jamming (JAM) upgrade. Thus the new baseline model includes communications jamming objects, and $R_b = 130\%, D_b = 108\%$. Reevaluating the remaining choices with the new baseline, we observe

X^2_{R+F}	1.13
X^2_{PER}	0.33

We chose the radio failure and repairs (R+F) to create a new baseline model. As a result, $R_b = 137\%, D_b = 112\%$ (recall that R_b is additive and D_b is multiplicative). At this point, adding task perishability would violate the constraint on R_b, so we're finished.

4 ISSUES OF PRACTICE: SHORTCOMINGS AND PROMISE

Given the evidence of the utility of adopting the standards of open architecture and object-oriented simulation, sponsors should support the reimplementation of models to position them for future growth. This step must be taken before any of the above analysis gains any degree of validity.

The methodology we describe here is clearly developmental and needs much improvement. We recognize the following as the major shortcomings:

- Data Risk Costing
  - risk weights are rough-hewn and arbitrarily chosen;
  - determining sensitivity of the model to data is difficult and time-consuming;
- Submodel Benefits
  - interactions of upgrades not accounted for;
  - constraint right-hand sides not objective;
- Evaluation Process
  - constrained optimization not the only approach;
  - greedy algorithm (used above) not guarenteed to produce optimal solution in this discrete optimization.

It would be extremely naive to propose that program managers construct the constraints and benefit functions as in the previous sections, and plow forward with the upgrade policy generated by the implied constrained optimization problem. However, pursuing the evaluation method described above will force development teams to compare the impacts of increasing resolution of submodels in a consistant manner.

Alternatively, suppose that a group of upgrades have been proposed by the organization responsible for the growth of a model. Oversight of these decision makers should involve examination of the analysis behind the selections, and possibly a formulation similar to that described here. It should be easy to verify that the selection of upgrades was made under reasonable assumptions, and that the constraints generated use reasonable models for performance loss, data source risk, and user or developer sophistication.

As has been illustrated in the example of MCCAAM, this framework for considering model fidelity is serviceable in the abstract, as well as when used on an existing model. Model developers should be obligated to produce similar reasoning whenever presenting plans for upgrades.

Finally, most developers realize that their job includes advocating model growth. This framework gives them a quantitative tool for arguing their case for model enhancement with program managers.

ACKNOWLEDGEMENTS

This work is an outgrowth of discussions with C^4I working group within the department of operations research at the Naval Postgraduate School. This group included Mike Sovereign, Chuck Chase, and Mike West.

REFERENCES

Agresti, A. 1990. *Categorical Data Analysis*, New York: John Wiley.

Bailey, M., W. Kemple, M. Sovereign, M. West, C. Chase, and C. Reece. 1991. An Object-Oriented Model of Communications for the MAGTF, *Proceedings of the Summer Simulation Conference*, ed. D. Pace, 472-481. Society for Computer Simulation, Baltimore, Maryland.

Balci, O. 1990. Guidelines for Successful Simulation Studies. in *Proceedings of the 1990 Winter Simulation Conference*, ed. O. Balci, R. Sadowski, and R. Nance, 25-32. IEEE, San Francisco, California.

Birtwistle, G. M., O. J. Dahl, B. Myhrhaug, and K. Nygard. 1973. *SIMULA Begin*, Philadelphia: Aurbach.

Cogliano, J. V. and L. W. Schruben. 1987. An Experimental Procedure for Simulation Response Surface Model Identification, *Communications of the ACM* Vol. **30**, No. 8, pgs. 716-730.

Glasserman, P. 1990. *Gradient Estimation Via Perturbation Analysis*, Boston: Kluwer Academic Publications.

Kleijnen, J. P. C. 1975. *Statistical Techniques in Simulation*. New York: Marcel Dekker.

Rothenberg, J. 1989. Knowledge-Based Simulation: An Interim Report, *RAND Note N-2897-DARPA*, The Rand Corperation, Santa Monica, California.

AUTHOR BIOGRAPHIES

MICHAEL BAILEY is an assistant professor of operations research at the Naval Postgraduate School. He actively pursues research in the fields of simulation modeling, experimental design for simulations, and applied stochastic analysis. He also has expertise in in several military applications areas, such as electronic warfare, C^4I, weapon system reliability, and transportation.

WILLIAM KEMPLE served for twenty years in the Marine Corps, gaining experience in communications and electronic systems, project management, research and development, computer-aided technical education, and command and control systems. He is currently an assistant professor of operations research at the Naval postgraduate School. His research interests include military applications of statistics and operations research, especially measuring the effectiveness of command and control systems.