On Post-Evaluation Analysis: Candle-Lighting and Surrogate Models

Author(s): Steven O. Kimbrough, Jim R. Oliver and Clark W. Pritchett

Reviewed work(s):

Source: Interfaces, Vol. 23, No. 3 (May - Jun., 1993), pp. 17-28

Published by: INFORMS

Stable URL: http://www.jstor.org/stable/25061745

Accessed: 01/03/2013 12:22

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at http://www.jstor.org/page/info/about/policies/terms.jsp

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.
Gleaning information from a model to guide the design of new and better options is an important, underemphasized facet of post-evaluation analysis. We call this facet candle-lighting analysis. We structure this analysis as a series of questions which we incorporate into a DSS that uses standard mathematical and artificial intelligence techniques. Creating new options typically requires action by an organization. The DSS stores models of these actions, their cost and benefits, and information about the source and accuracy of parameters. We applied candle-lighting analysis to a performance evaluation model developed with and for the US Coast Guard that is part of a prototype DSS.

The Coast Guard’s task of replacing its current fleet of ships with a new fleet, which it will use to pursue its missions for the next 20 or more years, is daunting in its complexity. The missions it will be asked to perform, the requirements to be met in performing those missions, the technology available for supporting its activities, and much else are known only in the most approximate ways. Still, important subproblems can be specified with sufficient clarity to be analyzed precisely, and hence DSSs can be used (and are being used) effectively on the fleet design problem.

We have been working since 1986 to design, implement, deliver, and support DSSs for the US Coast Guard and have sought throughout this project to extend the capabilities of DSSs [Kimbrough et al. 1990].
supporting acquisition and planning decisions, we have been struck by the need to have machine-based support for finding alternatives that are better than those assumed to be available. Often, the assumptions reflect what is currently feasible, yet with further investment such assumptions may be relaxed. For example, the effectiveness of a Coast Guard patrol is affected by

Often, the assumptions reflect what is currently feasible.

a number of factors, including the patrol vessel's speed, the type of radar it uses to spot other vessels, and its operational procedures. Having made reasonable assumptions about such factors, how can we go about finding promising opportunities to improve, for example, the speed of the vessel or the range of its radar? How can we best provide DSS support when the number of relevant factors is large? These were the principal questions that motivated our work.

Suppose we have built a model in a DSS, obtained data for the model, executed the model, and gotten sensible results for the model evaluation. It is universally recognized that at this point there is still much to be done with the model and much that a DSS can do to help. What is done at this point is often called *post-evaluation analysis* or *post-optimality analysis*. It may be thought of as (at least) three interrelated activities.

The first activity, which is commonly supported by DSSs, is to try to find out how stable the model is, given the data at hand. The technique used to answer this question, which we call the robustness question, is sensitivity analysis. We are interested in understanding how the model behaves under small perturbations of its assumptions. The simplex method for linear programming is an example of an algorithm that automatically provides a sensitivity measure in the form of shadow prices. This is unusual in that the sensitivity information is a byproduct of the simplex algorithm itself; for most other types of models, additional effort is required to obtain this information. Sensitivity information, however generated, can be used in several ways. It can be used both to measure the model's stability or robustness and to find out which parameters the model is most sensitive to so that we are directed to estimate those values with more precision. But beyond telling us what parameters to worry about, sensitivity information also illuminates opportunities by suggesting where to search most productively for better options.

The second component of post-evaluation analysis relates to a class of questions that we call structural analysis questions because answers to this class have normally been sought by systematically altering or manipulating the structure of the model in question. Two examples of questions from this class are: Why does the model give the answers it does? and What is it telling us about the system being modeled? We offer these two examples, but the questions in this group have not, so far as we know, been named in any widely accepted way. Much recent research work in model management has been directed at providing systematic support for structural analysis of models [Geoffrion 1987; Greenberg...
CANDLE-LIGHTING

1983, 1987a, 1987b; Greenberg and Murphy 1989]. This work has focused on optimization models but is for the most part not yet available commercially.

We are interested mainly in DSS support for the third component of post-evaluation analysis:

What can we learn from the model (which evaluates a given set of options) that will help us to find or design new and better options?

The more specific questions of this facet of post-evaluation analysis have not been named either. We call them candle-lighting analysis questions, after the motto of the Christopher Society: "It is better to light one candle than curse the darkness." By articulating the questions of candle-lighting analysis, we can begin to provide structured and automated support for answering them within a DSS. We envision a DSS that specifically supports designing new and better options, possibly ones that were not yet thought of, and then helps the decision maker argue for a course of action.

Such a DSS would support a variety of candle-lighting activities. However, this important facet of post-evaluation is not always emphasized. In one standard textbook, for example, Hillier and Lieberman [1986] offer the following perspective on post-evaluation analysis. They write that it is often

... implied that an operations research study seeks to find only one solution, which may or may not be required to be optimal. In fact, this usually is not the case. An optimal solution for the original model may be far from ideal for the real problem. Therefore, post-optimality analysis is a very important part of most operations research studies. ... This involves conducting sensitivity analysis to determine which input parameters are most critical in determining the solution, and therefore require more careful estimation, as well as to seek a solution that remains particularly good one over the entire range of likely values of these critical parameters. ... post-optimality analysis also involves obtaining a sequence of improving approximations to the ideal course of action (p. 22).

In other words, a given model, which embodies many assumptions, should be subjected to a post-evaluation analysis so as to understand how the results from the model depend on assumptions about the data and the structure of the model. Of particular interest here, Hillier and Lieberman mention generating new solutions but do not give details on how this is done. Generating plausible new solutions—for any type of model—requires thoughtful analysis of answers to certain questions.

Examples of Candle-Lighting Analysis

We will explain candle-lighting analysis with the aid of an example, a performance evaluation model we developed with and for the US Coast Guard, the barrier patrol model. One of the many missions the Coast Guard performs is monitoring and preserving the integrity of the US maritime borders. The barrier patrol model uses as a measure of effectiveness the probability that a randomly selected target vessel, attempting to cross a Coast Guard patrol barrier, is interceptable. The measure of effectiveness is,

\[ P(I) = P(I|D)P(D|A)P(A|O)P(O) \]  

where,

\[ P(X|Y) = \text{the probability of event } X \text{ given event } Y \text{ has occurred}; \]

\[ I = A \text{ (randomly selected) vessel attempting to enter a patrol barrier is interceptable}; \]

May–June 1993
D = A target vessel is detected by a Coast Guard asset that is available and can pursue interdiction, (the term asset encompasses airplanes, helicopters, and many classes of ships);

A = The Coast Guard asset is available for detection and pursuit, that is, it is not otherwise engaged with activities such as inspecting an intercepted vessel; and

O = The Coast Guard asset is on scene at the patrol barrier. We assume the asset is patrolling, pursuing, inspecting, or escorting.

The main model is simply the product of the four probabilities in (1). Some of these probabilities are determined by specific submodels while others are estimated constants. The first expression, \( P(I | D) \), is the probability that a target vessel is interceptable given that it is detected. In general, a detected vessel may or may not be pursued, depending on, for example, its speed or the estimated interception time. At present, this parameter value is a constant that we estimate from existing Coast Guard data.

\( P(D | A) \) is the probability that a target vessel is detected given that the Coast Guard asset is available. We determine this value using a simple submodel (appendix). We assume the Coast Guard asset has a detection area of radius \( r \) patrols a barrier of thickness \( 2r \) and length \( l \), and target vessels try to cross perpendicularly through this barrier. In addition to the parameters \( r \) and \( l \), we assume the speed of the Coast Guard asset is \( v \) and the speed of the target vessel is \( u \) (Figure 1).

\( P(A | O) \) is the probability that the Coast Guard asset is available given it is on scene in the patrol barrier. This value is determined by a Markov submodel (appendix). We assume that an on-scene Coast Guard asset can be in one of four states: on patrol, chasing a target vessel, inspecting a target, or escorting a target. With further assumptions about transitions between these states, we derive the value \( P(A | O) \).

The final parameter of the main model is \( P(O) \), the probability the Coast Guard asset is on scene at the patrol barrier. Currently,

\[ u \]
\[ l \]
\[ v \]
\[ r \]

Figure 1: The barrier patrol model predicts the probability that a Coast Guard vessel will intercept a random, intruding vessel. We assume the Coast Guard vessel patrols a barrier of length \( l \) and has a radius of detection \( r \). The target vessel approaches the barrier with speed \( u \), heading “south,” while the Coast Guard vessel, located in the center of the circle, is traveling at a speed \( v \) to the “west.”

INTERFACES 23:3
this is a constant equal to 1, but as we develop the barrier patrol model further, we may replace the constant with a specific submodel.

**Candle-Lighting Examples**

Sensitivity information may indicate that increasing the radius of detection $r$ may affect the probability of interception. This information is obviously valuable, but it is only part of what we need to make a decision. We may be able to improve the radius of detection, perhaps by installing more advanced radar or using a larger ship, but we should know about the costs and feasibility of such actions. The most obvious cost is purely financial, but such actions would also take time and might affect personnel requirements. Models of these costs allow us to make better decisions.

If the model is particularly sensitive to the value for the radius of detection and its value is uncertain, we should perform a simulation or some other analysis to understand how the model behaves for a range or a particular distribution of possible values. The results may suggest that we estimate this parameter more carefully or take action to decrease its variance. In the latter case, it would be valuable to have another submodel that explains how the radius of detection depends on the maintenance history of the radar or other factors that we may be able to control. In candle-lighting, we are primarily interested in providing the user with information that suggests what action to take to improve a parameter value.

Candle-lighting analysis is routinely performed for LP models and is greatly facilitated by the simplex method, which automatically provides useful sensitivity information. For example, the shadow prices properly interpreted guide us toward those right-hand-side constraints that can be most productively relaxed, allowing us proactively to explore new options. It is a way of lighting candles. However, if a parameter value, such as a right-hand-side constraint or an objective function coefficient, is determined by a submodel, the analyst must expend additional effort to interpret the sensitivity information. We would like to extend sensitivity information so that we automatically know the impact of changes in the lower level parameters. In this case, there is much that a DSS can do—and that the simplex algorithm does not do—to assist an analyst.

In general, a decision maker wants to know if the input parameters are derived from submodels, what these submodels are, what variables they include, how the overall result of the model changes based on ranges for these variables, and what action is required to change a parameter

---

It is better to light one candle than curse the darkness.

---

value. We maintain that this knowledge is a logical and frequently used basis of effective post-evaluation analysis, for which machine support is appropriate and needed.

Candle-lighting analysis is different from what-if analysis. What-if analyses examine the results of the model under different assumptions. There are many reasons for performing these analyses. Candle-lighting analyses are intended to help us in design-
ing new and better options. What is traditionally missing in DSSs is guidance in asking the right what-if questions; for the most part, the assumption is that the user has the intelligence to ask the right questions. Schoemaker [1991] warns, "What-if exercises are only useful if the right questions are asked . . . Giving someone a computer and spreadsheet without an appropriate thinking framework is not likely to be of great value." We cannot put all the appropriate intelligence of the analyst and the knowledge of the support staff into the system. Even so, we want the system to provide information that suggests appropriate courses of action.

**Surrogate Modeling**

Parameters to a model might be more than just raw data. We find it useful to broadly distinguish two cases. In the first case, a parameter value is determined by a true submodel, such as $P(D|A)$ in the barrier patrol model. In the second case, the parameter value is given as a raw datum but also has associated with it a surrogate model. We can think of the difference between a real (sub)model and a surrogate model as follows. In a real model, relations among variables are precisely and purportedly completely stated. In $F = ma$, there are no other parameters; $m$ and $a$ precisely determine the value $F$, although applying the model to a specific situation may only be an approximation. In a surrogate model there is no pretense that the relation is exact or that the given parameters are exhaustive. The surrogate model represents a subjectively assessed guestimate of, a rule of thumb about, the relationship between the parameter in question and one or more other parameters. A surrogate model provides a "quick and dirty" model of what might influence the value of a parameter, which is otherwise given as a raw datum. If it is a poor man's model, a surrogate model still has many of the virtues and uses of a real submodel, and in the absence of a real (sub)model a surrogate model is often useful.

Surrogate models are useful for candle-lighting in several ways. First, a given set of data (parameter values) may just represent our best guess at the moment. If these values might change over time, surrogate models can capture the quantitative change. Second, surrogate models can capture information that impacts, constrains, or informs the design of new options. We want models of what values are possible, what action (in dollars, time, and so forth) is required to achieve a given value or reduce the uncertainty of a value to a specified amount, and what is the impact (in dollars, efficiency, and so forth) of our actions. This is the kind of comprehensive information we need to make better decisions.

**Candle-Lighting Questions**

The basic question of candle-lighting analysis is, What can we learn from the model (which evaluates a given set of options) so that we can find or design new and better options? More specific questions we might want to ask follow. We generated this list from a combination of our best judgment based on experience and a review of the modeling literature. Clearly, additions are possible. The first two questions have to do with how current parameter values are determined and the impacts of changes in those values. The third question concerns the validity of the values and
CANDLE-LIGHTING

how they may change over time. Fourth, we ask what action is required to change values and what the impacts are. Last, we address qualitative issues and miscellaneous information relevant to the model.

Question 1—Are any of the input parameters to the model created by submodels? Of these, which are real submodels and which are surrogate models? In the barrier patrol model, some of the parameters in the top level model are determined by submodels, for example, $P(D|A)$. This is important because we cannot manipulate $P(D|A)$ directly but can do so indirectly through lower level parameters, such as the radius of detection. Similarly in the case of linear programming, before we interpret shadow prices or attempt to change a parameter value, we must be aware of how those input values were obtained. If they were calculated, we want to see the exact formula used.

Question 2—Do any submodels have common variables? This question builds upon question 1. If multiple parameters are based on submodels (real or surrogate) that have a common variable, a change in that variable may have an avalanche effect from the common dependency. We are interested in the impact of specific changes in these types of variables. For example, in linear programming we want to know by how much a common variable can change before the basis changes and what the effect on the objective function will be. In a typical production LP problem, the objective function may include item production cost coefficients, which may in turn depend on a common variable wage rate. How much can the wage rate—which, in this example is not a parameter of the linear program itself—change before the basis changes? If the submodels that determine the item production costs are simply linear, a new LP model could be developed in which the cost of labor is a parameter. However, such an option may or may not be practical, and we deliberately ignore it. The goal here is to handle the more general case of nonlinear submodels so that reformulation of the type described above is impossible.

Machine support for answering this question should allow the user to select, say hourly wage, from a list of common variables, and the following information would be provided.

—Common Variable: hourly_wage,
—Data: hourly_wage = 15,
—Submodel: prod_cost_item 1 = 45 = 3 hours $\times$ hourly_wage, and
—Submodel: prod_cost_item 2 = 75 = 5 hours $\times$ hourly_wage.

In the current specification of the barrier patrol model, there are no common variables. However, a more detailed model may consider the common variable maximum fuel capacity. This parameter could affect both the length of the patrol barrier and the proportion of time that the asset is available for patrol.

Question 3—How accurate is the result of the model evaluation, given the assumed accuracy of the model’s assumptions? Input parameters to a model can be actual amounts, estimates, or the results of submodels. We want to know the precision of

May–June 1993
It is a way of lighting candles.

is detected. This parameter is estimated from existing Coast Guard data. If there is known possible error for that estimate, we would want to know its impact.

We also want to generalize the concept of parameter accuracy in the following sense. We might want to know over what time horizon a parameter value will be accurate. If we have knowledge or are willing to speculate about such dynamics, a surrogate model could represent this. At least two simple cases should be handled: the case of a discontinuity and the case of continuous change. An example of the former would be a change in hourly wages resulting from a new labor agreement that takes effect in six months.

Question 4—Assuming we can act to change the model's assumptions, what is required to do so? The first three questions address how parameter values are determined and what the effects of changes are. Changing the values of the model's input parameters to our advantage will typically require an investment. It would be valuable to have models of the costs of such actions. The most obvious cost is purely financial, but there are others of interest, such as cost measured in time. While "time is money," the conversion formulas are not always obvious. Further, there may be good reasons to separate the financial from the temporal costs. Another example of generalized cost is effectiveness, the output measure of the barrier patrol model. In the case of a larger model, of which the barrier patrol model is a part, we may want to say there is a cost in effectiveness that would result from a given change in a parameter value.

By using surrogate models for this information, we provide valuable information within the system to the decision maker. Such a structured approach also allows for more automated reporting of the costs and impacts of a proposed change.

In our prototype implementation, we focus on two types of surrogate models of costs. One is a fixed investment. For example, changing the radius of detection from one value to another costs a certain amount and takes a specific length of time. The second simple model is the linear model, where, for example, we invest a certain amount to change a parameter value by one percent, twice that to change it by two percent, and so forth.

Here is the general form of a model for costs of actions:

model_name(action(), cost_of_action(value, type), results(new_values))
Ex 1: barrier_patrol_model(action(refit new radars), cost_of_action((10000, dollars),(6, months)), results(equals(r,35))
Ex 2: barrier_patrol_model(action(increase asset speed),
CANDLE-LIGHTING

cost_of_action((1000, dollars/day), result(increase(v, 10%))

Question 5—Who (or what) is responsible for the assumptions of the model and what about them might affect our action decisions? Information that is relevant to a model may be either quantitative or qualitative. A wide range of quantitative information can be captured in simple mathematical models. It is also important to capture the qualitative information that affects the validity of our model or its assumptions. For example, the radius of detection, r, may be based on an accumulation of subjective experience. It may prove important to know whose experience was relied upon and what the circumstances surrounding that experience were. Is r based on sightings in good weather or bad? How large were the vessels sighted? What were they made of, wood or steel? Other examples of issues that could have bearings on a model are possible actions of competitors, a change in the political environment, or other changes in states of the world. For now, we simply represent this information in free text form. However, we may wish to store some information as logical facts or rules, thus making it available for automated manipulation within the DSS. The example below shows a stylized way to store information about who is responsible for the model.

General form: developed__by(model__name, persons(person__list))

Example: developed__by(barrier__patrol__model, persons(Steve Kimbrough, Clark Pritchett))

Question 6—What else is there to know about the model or its assumptions that might guide us in taking action decisions? This general category is defined to hold everything else we want to remember about the model. Here we want to be able to store additional information that may be pertinent but simply is not stored anywhere else.

This is similar to having the ability to ask: Is there anything else I should know? What am I not asking? This could be very useful when a decision is being made by an individual not intimately familiar with the option (perhaps he or she didn’t actually gather the data).

This provides a way of adding information in a fairly unstructured and unanticipated way so that it can be automatically retrieved for display. However, it would not be available for automated inference, since its structure cannot be anticipated.

DSS Implementation

We implemented a prototype DSS that supports candle-lighting analyses and uses surrogate modeling. It is a fully functioning system with regard to the model computations and functions we describe.

The system has a HyperCard front end and uses Mathematica for mathematical operations that HyperCard is not equipped to do, such as generating normally distributed random numbers and raising matrices to large powers. (The communication between HyperCard and Mathematica is through file reading and writing, coordinated by a macro utility QuicKeys and the system scripting product Frontier. With a System 7 friendly version of Mathematica, direct inter-program communication can be handled more cleanly and easily.)

When an analyst begins a session with the system, she first chooses a model, in this case the barrier patrol model. The

May–June 1993
model is evaluated, and then the user can choose from a menu of post-evaluation options. Currently implemented are submodel, common-variables, parameter—
accuracy, cost—in—dollars, cost—in—time, responsibilities, and miscellaneous—
information.

If the user chooses submodel, she is presented with all the current submodels in the model, including the main model itself. If, for example, she chooses the main model, a summary section gives the value of the model is presented, and the equational form (1) is presented in a detail screen. A key feature of the system is the hypertext ability to click on any parameter in the model and be taken to its associated submodel. For example, if the user clicks on $P(D|A)$, she is taken to a similar screen in which a summary section gives the value of the parameter and the functional form of the submodel (appendix, eq. A1) is displayed in the detail screen. If the user continues to “drill down” [Minch 1989] into the model, the parameter values are ultimately given as raw data. At this level, the user is allowed to change the value of a parameter to a new value. The model is re-evaluated and the results of the change are reported. As an example, the user can change the radius of detection $r$ to a new value to see how such a change would percolate through the submodels to the main model.

When many parameter values are determined by submodels, some low level parameters may be common to several models. When the user selects the common—variables options from the post-evaluation menu, a list of all the variables common to more than one submodel is displayed. The user can select any parameter, change its value, and examine the results of the change. The barrier patrol model, in its present form, does not happen to have any common variables. However, this capability is operational, and a model in an earlier prototype made use of this capability.

Because parameter values are often uncertain, a simulation capability is available to explore the impact of variation in a parameter. When the parameter—accuracy option is selected, a list of all the parameters whose values are raw data is presented. For example, the radius of detection, $r$, would be on this list. If this is selected, the user is taken to a screen where the current value is given, and the user can enter five percent and 95 percent confidence interval values. A set of normally distributed random numbers for this range is generated by Mathematica. The model is repeatedly evaluated, and the mean and standard deviation of the repeated model evaluation is reported. Other capabilities, such as a histogram, could easily be added using Mathematica’s graphics capability and HyperCard’s picture display abilities.

The next two options use surrogate models of the costs associated with changing the assumptions of the model. Regardless of whether cost—in—dollars or cost—in—time is selected, a list of the low level parameters that have associated surrogate models is displayed. For example, if the user selects cost—in—dollars and then selects the radius of detection, $r$, a screen is displayed that shows the current value of $r$, along with a submodel giving the new value that results from installing new radar at a cost of $10,000. Similarly, for cost—in—time, a surrogate model shows that it
CANDLE-LIGHTING

would take a fixed time of 180 days to change \( r \) to any new value.

Two final options are available for the last two candle-lighting questions, responsibilities and miscellaneous information. When the user chooses the former she sees a simple, editable screen that explains who developed the model and who collected the data. In the current prototype this information is not stored as logical facts. We plan to add this capability in the next revision. Choosing the miscellaneous information option also presents a free-text screen.

Conclusion

The concepts of candle-lighting analysis and the technique of surrogate modeling are routinely used by modeling practitioners, but we have neither seen the concept applied systematically nor been able to find a literature on the subject, let alone an implementation described in the model management literature. So far as we are aware, ours is the first explicit discussion of candle-lighting analysis and of surrogate modeling in the context of DSS.

By carefully structuring candle-lighting analysis in a DSS, we are able to make such analyses easier and quicker to perform. This makes the DSS useful to a broader range of personnel than just specialists who have both knowledge of modeling and the details of the decision problem. In fact, rapid personnel turnover in the Coast Guard means that there may not be, at a given time, even a single individual who both understands the model and knows the relevant decision information we can store in surrogate models.

Although we explained candle-lighting analysis and surrogate modeling using the barrier patrol model, these concepts are general and can be applied to other models and systems. Our next step is to incorporate the candle-lighting analysis capability into other Coast Guard models. After that, application of machine learning algorithms for automated support of candle-lighting analysis will, we believe, prove an irresistible prospect.

APPENDIX

We will explain the two submodels used in the barrier patrol model more fully. \( P(D|A) \) is the probability that a target vessel is detected given that the Coast Guard asset is available. Based on the assumptions outlined earlier, and assuming that \( l > r \), we get as an approximation,

\[
P(D|A) = \frac{2r}{l} \left( 1 + \frac{v}{u} \right).
\] (A1)

This model can, of course, be made substantially more complex, but it is a useful approximation and will serve our purposes here.

\( P(A|O) \) is the probability that the Coast Guard asset is available given it is on scene in the patrol barrier; the states for the discrete time Markov submodel are

- \( S_1 \) = The Coast Guard asset is on patrol for target vessels,
- \( S_2 \) = The Coast Guard asset is attempting to intercept a target vessel,
- \( S_3 \) = The Coast Guard asset’s crew is on board, inspecting the target vessel, and
- \( S_4 \) = The Coast Guard asset has taken control of the target vessel and is either escorting it, guarding it, or (having released it) is returning to the patrol barrier.

The transition matrix is,

\[
P = \begin{bmatrix}
P_{1,1} & P_{1,2} & 0 & 0 \\
P_{2,1} & P_{2,2} & P_{2,3} & 0 \\
P_{3,1} & 0 & P_{3,3} & P_{3,4} \\
P_{4,1} & 0 & 0 & P_{4,4}
\end{bmatrix}
\] (A2)

May–June 1993
where \( P_{ij} \) is the probability of going from state \( S_i \) to state \( S_j \) in one time unit. With this formulation, \( P(A|O) \) is the value of the identical elements in the first column of the steady state transition matrix.

References


Kimbrough, Steven O.; Pritchett, Clark W.; Bieber, Michael P.; and Bhargava, Hemant K. 1990, "The Coast Guard’s KSS Project," Interfaces, Vol. 20, No. 6 (November-December), pp. 5–16.


R. A. Doughty, Chief, Coordination Branch, US Coast Guard, Research and Development Staff, 2100 Second Street SW, Washington, DC 20593-0001, writes, "The Coast Guard has been struggling for some time to get greater value from the many models we employ. We want to make them easier to use and more fully exploit their capabilities. The majority of our model users are not professional operation research analysts or computer scientists, but, they do know a great deal about the problem that a particular system is supposed to help them solve. The features presented in the paper on candle-lighting and surrogate models are a significant part of what our model users have been looking for. The idea of imbedding the post-processing capabilities in the modeling system to assist the user makes a great deal of sense. Having the computer system suggest how to change the variables to improve the output from a model is akin to having the model’s builder at the user’s side. Users now can spend more of their time addressing the problem that is at hand without being encumbered with the mechanics of the modeling system. The concepts presented in this paper are helping the Coast Guard give our model users more capability and independence."