

Just Modeling Through: A Rough Guide to Modeling

Michael Pidd

*Department of Management Science
The Management School
Lancaster University
Lancaster LA1 4YX
United Kingdom*

Skill in modeling is one of the keys to success in OR/MS practice. This has been recognized for many years, but we often give it only lip service. Models are used in many ways in OR/MS practice. A few simple principles of modeling may be useful. The six principles discussed here cover simplicity versus complexity; model development as a gradual, almost piecemeal process; dividing larger models into smaller components; using analogies; proper uses of data; and finally the way in which the modeling process can seem chaotic. Others may wish to comment on these principles and add their own.

When I am fortunate enough to visit a new country, I usually try to buy one of the *Rough Guides*, since the prejudices of the writers seem fairly close to my own. The guides point out the good (and bad) in the place to be visited, and they attempt the impossible by trying to give a flavor of the country in a few pages. I have been active in OR/MS both as an academic and as a practitioner since the early 1970s. In my experience, the real technical heart of OR/MS can be summa-

rized in the one word, *modeling*. In this paper, I will attempt to provide a rough guide to modeling, with principles that I and others have found useful and that seem to resonate with students and practitioners.

Others have written at length on the useful principles of modeling. Morris [1967] outlined some hypotheses about modeling that he had found useful and that illustrate the difference between modeling as an intuitive process and the for-

mal study of existing models. Little [1970] discussed how the then emerging technology of interactive computing could be employed to develop models that managers would be likely to use. Hugh Miser and Ed Quade had much to say on the subject in their magnum opus on craft issues in systems analysis [Miser and Quade 1988]. Hodges [1991] argued that even bad models may be used in satisfactory ways, even if those models fall short of their creator's original intentions. Pat Rivett has written much on the subject of modeling, and Rivett [1994] provided a number of examples to illustrate a general approach to model building. Powell [1995] discussed how modeling skills may be taught to MBA students and suggested six key modeling heuristics for this purpose.

I intend to complement the issues that these and others have raised. I also hope that I may stimulate other people to consider their own key aspects of modeling in OR. In discussing these principles, I will rely on my own background in discrete simulation. However, I believe the principles I discuss are relevant for most forms of mathematical and computer modeling in OR/MS. Some of the material in this paper is based on chapter 4 of my book [Pidd 1996].

Models and Modeling

I will focus on modeling as a verb or as an activity and not models as nouns or subjects. Morris [1967] wrote that "the teaching of modeling is not the same as the teaching of models." By models he meant approaches and methodologies, such as those of linear programming or queuing theory, that present ready-made models by which situations may be ana-

lyzed. In contrast to teaching such ready-made models, I, like Morris, am concerned with the processes of discovery and elaboration that are essential parts of modeling and of model development. I am interested in the ways people build and use models, rather than in the details of individual, ready-made models. Modeling activity is at the technical heart of OR/MS practice.

Models in OR/MS have two main uses: (1) People use models to explore the possible consequences of actions before they take them. An example of this is the evaluation of weapons systems in defense [Cannella, Sohn, and Pate 1997]. Boothroyd [1978] called this "reflection before action." Used in this way, a model is a convenient world in which one can attempt things without the possible dire consequences of action in the real world. In this sense, models become tools for thinking. This thinking might relate to one-time events, of which an example might be a particular capital-investment decision. Or the thinking might concern occasional events, such as pricing reviews. Alternatively, the thinking might concern routine events, as in weekly production planning. We also use models as tools for thinking when we try to understand a complex system, even if we contemplate no immediate action.

(2) The other broad use for models in OR/MS is as part of embedded computer systems for routine decision support. This use is very common in the management of logistics and supply chains [Arntzen et al. 1995]. When used in this way, the models form an essential and automatic part of the management and control of an organi-

zation. This does not mean that there will never be human intervention, but it does mean that the model or modeling system is usually intended to get things right.

I discuss models as tools for thinking, that is, as ad-hoc exploratory devices for reflection before action.

Models are no substitute for thought and deliberation.

With this in mind, I define a model in OR/MS as follows: A model is an external and explicit representation of part of reality as seen by the people who wish to use that model to understand, to change, to manage, and to control that part of reality in some way or other.

This definition has a number of important features. First, the model is external and explicit rather than existing as a set of mental constructs that are not accessible to other people. This means that the model can be examined, can be challenged, and can be written in a logical language, such as that of mathematics or of computer programming, and it may even serve as a form of organizational memory when a model-building team is in action.

Second, the model is a representation of the real world. Checkland [1995] distinguished between two types of model. First are those intended to be representations of the real world, and these are, I suspect, typical of those employed in OR/MS. This does not mean that the models are complete or as complex as the world they represent. It does, however, mean that they are intended to represent certain aspects of the real world. They then become surrogate forms of reality that can be safely ex-

plored and manipulated. Checkland's second type of model includes the conceptual models used in soft systems methodology (SSM). These models are intended to embody the elements that should be present in an idealization of a system under scrutiny [Checkland 1981]. Checkland's "should be" comes from systems theory and its notion that any viable system will include a number of essential activities, such as those needed for control by feedback. Thus a conceptual model, as used in SSM, serves as a basis for debate about the differences between the situation as it is now and the situation as it might become. In this sense, Checkland's conceptual model clearly need not be a representation of reality. I focus here on the first type of model, those that are would-be representations of the real world.

My definition's third feature is the assumption that no model as used in OR/MS will be a complete representation of reality. If it were, then it would be as complicated, as expensive, as hard to manipulate, and as disastrous when things go wrong as reality itself. Instead, the representation is partial, and the partiality is governed by people's intended use of the model. They want it to be fit for some purpose. These people want to use the model to understand, to change, to manage, and to control that part of reality in some way. Thus, the modeling is goal-oriented. This does not, however, imply that the participants necessarily agree about the intended use of the model.

The difficulty with assuming that models are representations of even part of reality is that people may differ in what they regard as reality or may disagree over

what part of reality to model. To appreciate this, one need only realize that models are always simple, whereas realities are always complex. An old mathematical joke about numbers helps to illustrate this.

Complex numbers have two parts—the real and the imaginary. A system being modeled is like a complex number, with real and imaginary parts. The problem is that different people see things in different ways; one person's reality looks like another's imagination, and vice versa. Thus a complex system may be impossible to model in all its complexity.

With these ideas in mind, I consider six simple principles of modeling:

- (1) Model simple; think complicated.
- (2) Be parsimonious; start small and add.
- (3) Divide and conquer; avoid megamodels.
- (4) Use metaphors, analogies, and similarities.
- (5) Do not fall in love with data.
- (6) Model building may feel like muddling through.

My own technical expertise in discrete simulation is bound to color my choice of principles, but I believe they are generally applicable.

Principle 1: Model Simple; Think Complicated

My definition of a model suggests that models are simple representations of complex things. How then can they be adequate devices to support reflection before action? A particular problem is the need for variety noted by Ashby [1956], of which one statement is that "variety must match variety." Ashby's principle of requisite variety stems from tenets of control theory, which state that a control system

must be able to match the full variety of the system that it is controlling if it is to be of much use. Thus, a device to control the temperature of a domestic furnace must be able to detect and act when the furnace gets too cold as well as when it is too hot. Ashby took this common-sense notion and developed it mathematically.

Does this mean that a model must be as complicated as the system it represents? Thankfully, no. Were the answer yes, modeling would be uneconomic, since a model would take as long to build as the system it represented—and it would be just as expensive to develop and control.

The model alone need not satisfy Ashby's principle; rather the system that comprises the model and the user(s) must match this variety. That is, in systems terms, the model:user(s) system displays emergent behavior that must match that of the system being modeled. This notion explains this first principle of modeling: model simple; think complicated. Models are no substitute for thought and deliberation; they are part of a process of reflection before action. That process must have requisite variety, not the model alone. Little [1970] made a similar argument for simplicity when considering how managers actually use models. He argued that people use models as a means of decision support and that the model:user system is a form of man:machine system more powerful than the human or the model alone.

This principle means that a relatively simple model can support complicated analysis. A model intended as a tool for thinking needs to be adequate for the task, and it must be skillfully used if the task is to be done well. It implies a mind-set dif-

ferent from that occasionally evident among OR/MS people. For example, at a conference I attended, a plenary speaker, a distinguished and senior manufacturing manager of one of the world's most successful companies, told the audience that he wanted a model to simulate all his company's worldwide factories at almost any level of detail. As he was our distinguished guest, it would have been rude to disagree. The appropriate response, though, is that given by Evelyn Waugh's character Boot to his boss, Lord Copper, in the novel *Scoop*. Whenever Lord Copper expressed some outrageous view, Boot's polite response was, "To a point, Lord Copper, to a point!" No model can or should do everything.

Why is simplicity desirable? Little [1970] argued that models should be transparent (that is, simple to understand, at least in outline form) and should be easy to manipulate and control. Transparency is desirable because successful OR/MS practice depends on trust between consultant and client. Trust is easier to establish if the client can appreciate the overall workings of the model and understand what the model can and cannot do and why. However, this notion of transparency does not imply that a management science model must be limited by the technical prowess of the people sponsoring the work.

Ease of manipulation is a goal that is much easier to satisfy now than it was for Little in 1970. We take interactive computer facilities for granted, and their use, especially with visual interactive approaches, can make decision models very easy to manipulate. This brings a danger in its train. A powerful model is like a

chain saw: used properly it is a very useful tool; used without training it can cause considerable damage, some of it to the user. The opposite is also true: a powerless but easily manipulated model (one that does not fit its intended purpose) is like a chain saw with no cutting chain. It makes a lot of noise but isn't of much use.

"To a point, Lord Copper, to a point!"

A simple model does not have to be a small model. In mathematical terms, simplicity can be regarded as the close relative of elegance. Knowing when to simplify requires considerable understanding of the system being modeled and of the tools of modeling. In wrestling with theological problems, William of Ockham, a British monk who lived around 1200 AD, argued that "a plurality (of reasons) should not be posited without necessity." This principle, "Ockham's razor," is an argument for simplicity of explanation, the parallel of simplicity in representation.

Simplicity can never be an end in itself. Sometimes complex models are needed, especially when they are to be part of embedded real-time systems. Most modern jetliners are at least partially automated and include avionics systems that control the aircraft. These systems and their models must meet Ashby's principle of requisite variety, otherwise automatic landing and flight would be impossible with any degree of safety. However, Little's notion of transparency still holds good: the pilot should understand how the model of the flight surfaces is intended to perform, even if he or she does not understand the

detailed theory.

Unfortunately we have no metric for simplicity, no way of knowing whether our models are optimally simple (to coin a strange term). Presumably we can put a crude lower bound on this optimal simplicity. For example, the mental models in use before the OR/MS modeling could serve as a lower bound, since we would otherwise have no reason to develop the formal OR/MS model.

How can we train students and ourselves to produce models that are just complicated enough for the task at hand? First we must be clear about the task at hand. If our models are to be fit for purpose, we must be sure what that purpose is. Problem structuring and framing is important [Smith 1988, 1989; Pidd and Woolley 1980]. To quote a maxim attributed to John Dewey, "a problem well put is half solved." Smith [1989] argued that problem definition is a form of representation, that is, a step beyond a mental model but before a formal OR/MS model. This being so, structuring a problem properly is a key to forming a lower bound on the optimal simplicity of a formal model.

Principle 2: Be Parsimonious; Start Small and Add

Given that it is impossible to know in advance how complicated a model has to be, it seems sensible to approach its construction like an army advancing on a well-defended city at night, that is, with a little stealth and cunning. In teaching computer-simulation methods, I always recommend the principle of parsimony: that one should develop models gradually, starting with simple assumptions and adding complications only as necessary.

Rather than attempting to build a final model from scratch in one single heroic effort, we make initial assumptions that we know are too simple, and we relax them later. The idea is that we will refine this initial far-too-simple model over time until it is good enough, until it is fit for its intended purpose.

This deliberate attempt to start small and add is Morris' [1967] first proposal or hypothesis about modeling. He argued that modeling should be viewed as a process of enrichment or elaboration in which the modeler moves from simple representations to more complex ones. The modeler elaborates the model as a result of "confrontation with data" or simplifies it if the representation is intractable (in the case of a mathematical model).

Miser and Quade [1988] provided an interesting example of this confrontation with data. They discuss the development of a model to investigate how much of an aircraft runway would be visible to pilots under different degrees of cloud cover. As an initial simplification, one could represent the clouds by circular disks. Clouds in the sky are not truly circular, but this simplification is attractive because the geometry of a circle is simple and because one could use several disks to represent more complex shapes. Whether this simple model is accurate enough can be assessed by real-world trials, which might show that more complex geometry is sometimes needed.

This parsimonious modeling can be approached through what I call prototyping, a term also used by Powell [1995] and similar to prototyping in software engineering. The idea is that analysts should

deliberately develop a series of models, each more complex than its predecessors. The initial models will be far too simple but will provide insight to incorporate more formally into later models. The modeler builds models that are too simple and, when their limitations become obvious, throws them away and builds another to get round some of the limitations. Thus, through a series of prototypes, the modeler gradually produces a model that is fit for its intended purpose. In *The Mythical Man-Month*, Brookes [1975] argued that software developers should be prepared to throw away their early attempts at designing a system. He insisted that they will end up discarding them anyway, so why not build this knowledge into their planning and allow time for it? Thanks to modern computer software, it is simpler to develop prototypes in this way than it would have been 20 years ago. Even if a model is discarded, it is sometimes easy to keep components and build them into the later, better models.

A second approach to this parsimonious modeling is to gradually refine simple models by checking whether they are adequate for their intended purpose. Sometimes the simple models are best scrapped, but through judicious use of modular approaches one can sometimes achieve a form of stepwise refinement. The modeler deliberately starts with a model that is too simple, but from which he or she can gain a few lessons and insights. The modeler then gradually refines the model, for example, by replacing deterministic elements by stochastic ones, each time checking to see if the model is now good enough for its intended purpose.

This is often possible in computer simulation projects if the modeler takes a deliberately modular approach by using an object-oriented methodology or by following a scheme such as DEVS [Zeigler 1976, 1984]. The idea is to divide a model into components that can be replaced if they are too simple, without the need to redevelop the entire model. Such an approach provides a modular structure that facilitates the testing and implementation of the model. Rather than discuss this idea in isolation, I will include it in my discussion of the third principle.

Principle 3: Divide and Conquer; Avoid Megamodels

Writing about a similar idea, Powell [1995] calls this decomposition. Raiffa [1982, p. 7] had this to say after developing large-scale models at IIASA:

Beware of general purpose, grandiose models that try to incorporate practically everything. Such models are difficult to validate, to interpret, to calibrate statistically, and, most importantly to explain. You may be better off not with one big model but with a set of simpler models.

Developing a set of simple models is perhaps most useful when a large model is needed, especially when a team of people is engaged in its development. Models can be developed as a set of interrelated component models, each of which can be understood and tested easily. Each component model should be developed with the full set of modeling principles in mind. This avoids overgeneral models that cannot be validated and provides a scheme within which the components can be validated as part of the whole.

As an example of this decomposition, consider a problem I came across in prac-

tice: A manufacturer of packaging items wished to provide better service to its geographically scattered customers. One measure of this service is the proportion of orders that are filled when ordered, on time, and in full. The graduate students who tackled this problem approached it by building three separate models:

- (1) They built a model of the demand for the main classes of product, showing the variation from day to day and by season and the growth trends. In essence, this was a standard time-series model.
- (2) They built a model of the production capacities for the range of products, allowing for different product mixes of the materials (for example, bubble wrap) that form the items the manufacturer sells.
- (3) They built a model of the process by which stock may be allocated to customers, which depends on the priorities placed on particular orders and on the likely sizes of trailers available for distribution.

Each of the component models could be separately tested and tweaked. In addition, the students could discuss each model with the people who ran the system and satisfy them that the model incorporated enough detail. By combining these models, the students could show what actions the manufacturer might take to improve customer service.

When operating in this way, the analyst should be aware of system effects. One fundamental idea from general systems theory is that of holism, roughly defined as, the whole is more than the sum of its parts, and the part is more than a fraction of the whole.

A system may not work even though it

is built from components that individually function well. The interaction of the components, their links, causes the problems. This is a well-known problem in discrete simulation modeling that occurs when models are developed from the building blocks of events, activities, and processes. The risk is that the component models embody subtly different assumptions, which may lead to very strange behavior when they are linked together. As a simple example, a manufacturing company may have customers who do not place orders on public holidays, even though the manufacturer produces on those days. If the modeler does not realize this, models of demand and of production capacity, when combined, may lead to wrong conclusions.

“A problem well put is half solved.”

Bizarre though it may seem, one advantage of this principle of model decomposition is that a model that is invalid for one purpose may be a valid component of a system of models for some other purpose. Hodges [1991] mentioned this when discussing “Six (or so) things you can do with a bad model.” He suggests that a bad model (one that is either invalidated or unvalidated) may become a satisfactory part of an automated management system if it is treated as a “dumb-looking transparent box.” By this he means that the model, within the range of behavior required by the overall system in which it is embedded, may produce satisfactory outputs from the inputs that it receives. This may be especially true of time-series models that are valid across a very limited

range of data.

The successful use of this principle of divide and rule seems to depend on ensuring that assumptions are well documented and that all those doing the modeling understand these assumptions.

Beyond the assumptions underlying models, analysts should be aware of such factors as the level of detail in each component model and the compatibility of the interfaces between these models.

Principle 4: Use Metaphors, Analogies, and Similarities

Morris [1976] recommended that modelers seek an analogy with some other system or an association with some earlier work. In seeking such an association, the modeler relies on his or her previous experience or stands on the shoulders of those who have gone before. In either case, the idea is to search for previous well-developed logical structures similar to the problem at hand.

In their 1960s text on OR, Ackoff and Sasieni [1968] suggested that analog models are sometimes used in OR. By an analog model, they meant a model in which one property is replaced by another that is easier to manipulate, and they gave a number of examples. I suspect that direct analog models of the type to which they refer are rarely used in OR/MS practice. However, metaphors and analogies are certainly useful in modeling. In one sense, any mathematical model is an analog in which a physical property or management policy is replaced by a mathematical representation. However, analogy usually implies more than this.

Axelrod [1976] and Eden, Jones, and Sims [1983] gave an interesting example of

an analog model that is not directly mathematical. Both suggest the use of cognitive maps to capture the links in people's arguments when they discuss possible decisions. A cognitive map is a directed graph in which the nodes represent concepts and the directed arcs represent relationships of means to ends expressed by individuals or groups. Thus the arcs represent the psychological relationships between concepts, that is, they serve as analogs for the organization of the mental constructs that characterize the thinking of the individual or group.

Analogies are also useful in modeling as an activity. A common but misconceived view of creativity is that it is the ability to conjure new ideas, apparently from nowhere. In practice, much creativity comes from realizing that ideas and notions developed gradually in one field can be usefully applied in another, that is, from seeing links that others have ignored or could not see. Saaty [1998] described these as deductive creativity. Saaty, however, was pleading for a new paradigm of OR/MS in which analysts employ what he termed inductive creativity, a deliberate attempt to look at experience and to induce a larger and more general system view of what is happening. They thus see a particular problem as merely an instance of a general case.

Evans [1991] was particularly concerned with the development of creativity in OR/MS, and like Morris [1976], he advised the use of metaphor and analogy. A metaphor is a figure of speech in which one thing serves as a surrogate for something else that it resembles. Thus, we might speak of a problem as being a hard

nut to crack. We draw an analogy when we point out the agreement or correspondence in certain respects between two different things. In OR/MS modeling, we might deliberately draw an analogy between, say, the demands on an emergency service and demands in a queuing system. The two are not the same, though both may be regarded as having servers and customers, and they have enough similarities that we can transfer learning from one to another. This use of analogy is close to the notion of model enhancement by association [Morris 1967].

Synecotics, developed at Arthur D. Little and described by Evans [1991], is a more general attempt to use analogies by encouraging participants to draw different types of associations in tackling an issue. Synectics suggests three analogies relevant to OR/MS:

- (1) In personal analogy, participants imagine themselves inside the systems being modeled. This is a very common approach in discrete simulation in which modelers may try to imagine the states through which important system entities pass. For example, they might imagine themselves as cars finding their way through a congested road network via a sequence of junctions.
- (2) Direct analogy is the type drawn above between an emergency service and a straightforward queuing system. The analogy need not be perfect (indeed it could be argued that it can never be perfect), but the idea is to transfer lessons learned in one sphere to another.
- (3) Fantasies, which are not really analogies, permit participants to stretch the boundaries and imagine the occurrence of

something that may be currently infeasible as a way of directing attention toward the important features of the issues under consideration. An example might be something like, "OK, let's suppose that we have some way of instantaneously making any blood type available. What then would be the important features of a blood-transfusion service?"

It seems likely that analogies are most useful in the initial stages of modeling [Morris 1976]. They can be used to illuminate the development of the initial simple models suggested by the principle of parsimony.

Principle 5: Do Not Fall in Love with Data

The availability of friendly computer packages has produced a generation of people who are hooked on data, data junkies. I am not arguing that data and its analysis are unimportant and can be ignored. Rather, I observe that many students seem to imagine that modeling is impossible without data, preferably lots of it. They treat data as a form of Linus' blanket. They assume that, because a model is a representation of some system, examination of data from that system will reveal all that they need to construct the model. Such an assumption may be a mistake, even though exploratory data analysis is very useful, and I would not wish to return to the days before modern computer software appeared for this purpose. The slapdash use of exploratory data analysis can never replace careful thought and analysis.

Modeling should drive any data collection and not the other way around. The modeler should think about the type of

model that might be needed before attempting large-scale data collection.

Conway et al. [1995, p. xx] gave some very sound advice about the use of data in the discrete simulation of manufacturing systems, including the following assertion:

You should resist the temptation, or the instructions, to collect real data with all the ingenuity you can muster. The collection process is much less valuable than you might think.

Although this is an extreme view, it is a useful counter to the familiar greeting of the naive modeler to his client, "Take me to your data." Only after thinking about the model can the analyst know what type of data to collect.

Go for an appetizer before the main course

It can be very helpful to recognize that a modeling exercise may require two types of data, both of which may be qualitative as well as quantitative. Preliminary data, collected early in a modeling project, form part of the problem structuring, during which the modeler frames and names the important issues. The modeler uses preliminary data to get an idea of what type of model is needed.

Being British, I often remind my OR/MS students of Rudyard Kipling's verse from the *Just So Stories*:

I keep six honest working men
(They taught me all I knew);
Their names are What and Why and When
And How and Where and Who.

Much loved by industrial engineers under the heading of critical examination, the verse suggests six questions that are important in structuring problems. Their

answers form the preliminary data, which is the appetizer before the main course. I also call these the six idiot questions, since they can be asked only in the early stages of a project. Ask them too late, and your client thinks you are an idiot.

The verse also illustrates the important point that preliminary data are not just quantitative; the qualitative is important in framing the issues to address. This is not to say that a preliminary quantitative analysis is a waste of time, but numbers are not enough.

Beware of data brought on a plate—order à la carte, not table d'hôte

With modern information systems, many organizations are awash with data, but it is worth remembering that old adage of IS: information is data plus interpretation. When taking data straight from a corporate IS, one risks misinterpreting it. In this way, MIS information becomes misinformation. In today's global organizations, many people work in virtual teams and this issue becomes particularly important. Just like liquids to be pumped, data put into an MIS are usually filtered and cleaned up. This filtering and cleaning may have such an effect that the data are of little use in analysis and modeling.

As an example, suppose a hospital wished to reduce its waiting lists and was proposing to model some aspects of its operation in order to decide whether to do so. One might think that the best way to obtain data about patient referrals to the hospital would be to examine the referral database that contains information about actual referrals. However, this data might be very misleading for a number of reasons. First, doctors who refer patients to

the hospital may have a choice of hospitals and may know how long their waiting lists are. They may refer their patients to hospitals with short lists. Slightly more subtly, they may refer them to hospitals that they believe have short lists. Further, the waiting lists are a symptom of a problem, not the problem itself. As doctors can tell you, they give symptomatic or palliative treatment only when there are no other options; if possible they treat the underlying causes. Thus, it may be better to treat the waiting lists as outputs from a model rather than as inputs.

Eat healthily—avoid anorexia and gluttony

People on a starvation or subsistence diet must find it perverse that so many people in modern Western society have eating disorders. Two such disorders are anorexia, a form of deliberate self-starvation, usually because of a false body image and low self-esteem, and gluttony in which excessive food intake leads to obesity. Without stretching this food analogy too far, there are clear parallels in the collection and use of data for OR/MS modeling.

The equivalent of anorexia occurs when analysts or groups decide that they know best. Although I am convinced that the model should drive the data collection and not vice versa, sometimes this can be dangerous. We can be tempted to disengage from the world in which our clients and customers work. We may feel that we know best: "We've done so much work in this area that we can pretty easily translate previous projects into something workable here." Sadly, this false image of our own prowess may be exposed only when the

analysis dies of starvation and it is too late for force feeding. The other reason for such apparent arrogance may be, paradoxically, that we feel inadequate in the arena in which we are invited to work. Thus, we shy away from engagement and end up without enough data of the type we need.

Numbers are not enough.

An equivalent of gluttony is seen as an obsession with data, especially in huge quantities. Sad to say, modern computer systems may encourage this. For example, it is possible to work through huge data sets even if they are geographically dispersed. Because this can be a fascinating thing to do, it may become something we wish to do. But the real question to ask is, "Is it necessary?" It may be better to work with a single detailed data set and then take samples from others to check that the detailed set is representative. Careful planning and use of data are as important in OR/MS as are exercise and a balanced diet to us as people. It may even be a good idea to have a data fast, much as suggested in the earlier quote from Conway et al. [1995].

Other foods are available—if you ask

In collecting and analyzing data, one must remember that any data are just samples from some population. When asked for a forecast, cynical economists advise a golden rule, "Give a number or a date, but never both!" The data used in modeling should always be dated. When we use data to build or to test a model, that data will have been collected at a particular time and over a certain period. If

we say the data are believed to be representative, we imply that the data are representative of a larger population that displays some regularity through time. We do not expect someone to surprise us later with drastically different data. Nevertheless this may happen, and it is always a risk, especially when we use the results of a model to extrapolate into the future. The future may simply differ from the past, and the population from which the data sample comes may behave differently in the future.

Data also form a set of observations. The modeler must realize that data are samples of what he or she might obtain given enough time and other resources. Any observation process is subject to errors of different types. It is reasonable to be skeptical about the resulting data, especially if it is readily available. In a persuasive example, Morse [1986] described the early operations research efforts in wartime, including this quote from a pilot who had been asked to provide reports after each mission. "Hell, I didn't think anyone ever read those damned reports!" On hearing this, Morse and his team chose to collect their own data.

Principle 6: Modeling May Feel Like Muddling Through

Muddling through is a very British concept that seems to go with the idea of the gifted amateur. It is also a fairly accurate description of how it feels to be build models in OR/MS. It is tempting to see model building as a linear process in which we move from step 1 to step 2 to step 3 and so on. We might concede that some people may need an extra step 2a, but we often carry this simple view of

modeling in our minds. The reality is very different. As a parallel, consider the process of PhD research and the final thesis. It seems unlikely that most such theses describe how the student actually spent his or her time during the years of research. Instead, the student presents a summary within a strictly defined logical framework. This presents a problem for many PhD students, since they have to place rational constructions on a process that was probably shot through with intuition, hope, and despair. A pretense that modeling is a rational and linear process may create similar problems, especially for newcomers.

Willemain [1994, 1995] set out to find how expert modelers actually work and found that they rarely operated like Mr. Spock in *Star Trek*. His work was in two parts, both with experienced OR/MS practitioners and academics. In the first stage, he interviewed them to ask them to give accounts of how they went about their modeling work. Interestingly enough, their accounts had much in common, and they claimed the following. They [Willemain 1994, p. 214] "develop their models, not in one burst, but over an extended period of time marked by heavy client contact." Also, they are "guided by analogies, drawing and doodling, they develop more than one alternative model, in each case starting small and adding." Many of their claimed approaches follow the principles I have discussed.

In the second stage of his research, Willemain placed the modelers individually in a room, set a modeling task, and asked them to try to develop a suitable model, all the time talking through what

they were doing. There can be no certainty that their talking accurately reflected their thinking; however, analysis of their talking revealed some interesting things.

The tapes show that the modelers devoted about 60 percent of their time to model structure, that is, what we would regard as the core of model building. They divided 30 percent of the time equally between problem context and model assessment. They devoted just 10 percent to model realization and almost none to implementation.

In the study, the modelers had just 60 minutes to work on a problem, so it is not surprising that they devoted so little time to model realization or implementation. But what is significant is that they spent so much time thinking about problem context and model assessment. What is also very important is that the time they spent on the three major concerns was scattered through the modeling session. They kept picking up a concern for a while, dropping it, and then returning to it. Presumably this pattern would be even more marked were it possible to follow how they operate over a much longer period in their real work. Muddling through is quite a good description of how these experienced people went about their work.

An Invitation

Like any book, this "rough guide" must come to an end. Like all authors of such tourist guides, I can describe only the main features of the land being visited. Most such authors exhort their readers to send in comments and corrections. I do too.

Many writers have proposed principles of modeling, all of which are a welcome

relief from justifications of why OR/MS students must know about the various linear-optimization algorithms. Other people will probably have their own lists that differ from mine and the others I've mentioned. I would like to hear from them. I hope we can embody these and other principles in the courses we teach. Students need to realize that learning the skills of modeling may be more important than learning about models.

This paper is also a plea for some serious research about how people go about their modeling. Tom Willemain made a start in this area. With Bob Woolley, I did some small-scale empirical work in the late 1970s, but much has changed since then, both in the organizational world and in the tools available to support model development. Smith [1989] proposed a prescriptive framework, based on cognitive psychology as well as on OR/MS, which makes some sense of modeling as an activity. Perhaps someone could take some of these ideas and conduct more empirical research to see if there are better ways of doing things.

References

- Ackoff, R. L. and Sasieni, M. W. 1968, *Fundamentals of Operations Research*, John Wiley and Sons, New York.
- Arntzen, B. C.; Brown, G. G.; Harrison, T. P.; and Trafton, L. L. 1995, "Global supply chain management and Digital Equipment Corporation," *Interfaces*, Vol. 25, No. 1, pp. 69-93.
- Ashby, R. 1956, *An Introduction to Cybernetics*, Chapman and Hall, London.
- Axelrod, R. 1976, *Structure of Decision: The Cognitive Maps of Political Elites*, Princeton University Press, Princeton, New Jersey.
- Boothroyd, H. A. 1978, *Articulate Intervention*, Taylor and Francis and the Operational Research Society, London.
- Brookes, F. P. 1975, *The Mythical Man-Month*:

- Essays on Software Engineering*, Addison Wesley, Reading, Massachusetts.
- Cannella, D. A.; Sohn, S. Y.; and Pate, C. 1997, "Evaluating new weapon systems technology: The Javelin versus the Dragon," *Interfaces*, Vol. 27, No. 3, pp. 29–38.
- Checkland, P. B. 1981, *Systems Thinking, Systems Practice*, John Wiley and Sons, Ltd, Chichester, England.
- Checkland, P. B. 1995, "Model validation in soft systems practice," *Systems Research*, Vol. 12, No. 1, pp. 47–54.
- Conway, R.; Maxwell, W. L.; McClain, J. O.; and Worowa, S. L. 1995, *User's Guide to XCELL Plus: Factory Modeling System*, Release 4.0., Scientific Press, Palo Alto, California.
- Eden, C. L.; Jones, S.; and Sims, D. 1983, *Messing About in Problems*, Pergamon Press, Oxford, England.
- Evans, J. R. 1991, *Creative Problem Solving in the Decision and Management Sciences*, South Western Publishing Company, Cincinnati, Ohio.
- Hodges, J. S. 1991, "Six (or so) things you can do with a bad model," *Operations Research*, Vol. 39, No. 3, pp. 355–365.
- Little, J. D. C. 1970, "Managers and models: The concept of a decision calculus," *Management Science*, Vol. 16, No. 8, pp. B466–485.
- Miser, H. J. and Quade, E. S. 1988, *Handbook of Systems Analysis: Craft Issues and Procedural Choices*, John Wiley and Sons Ltd, Chichester, England.
- Morris, W. T. 1967, "On the art of modeling," *Management Science*, Vol. 13, No. 12, pp. B707–717.
- Morse, P. M. 1986, "The beginning of operations research in the United States," *Operations Research*, Vol. 34, No. 1, pp. 10–17.
- Pidd, M. 1996, *Tools for Thinking: Modelling in Management Science*, John Wiley and Sons, Ltd, Chichester, England.
- Pidd, M. and Woolley, R. N. 1980, "Four views on problem structuring," *Interfaces*, Vol. 10, No. 1, pp. 51–54.
- Powell, S. G. 1995, "The teacher's forum: Six key modeling heuristics," *Interfaces*, Vol. 25, No. 4, pp. 114–125.
- Raiffa, H. 1982, "Policy analysis: A checklist of concerns," PP-82-2. International Institute for Applied Systems Analysis, Laxenburg, Austria.
- Rivett, B. H. P. 1994, *The Craft of Decision Modeling*, John Wiley and Sons, Ltd, Chichester, England.
- Rosenhead, J. V., ed. 1989, *Rational Analysis for a Problematic World*, John Wiley and Sons, Ltd, Chichester, England.
- Saaty, T. L. 1998, "Reflections and projection's on creativity in operations research and management science: A pressing need for a shifting paradigm," *Operations Research*, Vol. 46, No. 1, pp. 9–16.
- Smith, G. F. 1988, "Towards a heuristic theory of problem structuring," *Management Science*, Vol. 34, No. 12, pp. 1489–1506.
- Smith, G. F. 1989, "Defining managerial problems: A framework for prescriptive theorizing," *Management Science*, Vol. 35, No. 8, pp. 963–981.
- Willemain, T. R. 1994, "Insights on modeling from a dozen experts," *Operations Research*, Vol. 42, No. 2, pp. 213–222.
- Willemain, T. R. 1995, "Model formulation: What experts think about and when," *Operations Research*, Vol. 43, No. 6, pp. 916–932.
- Zeigler, B. P. 1976, *Theory of Modeling and Simulation*, John Wiley and Sons, Inc., New York.
- Zeigler, B. P. 1984, *Multi-facetted Modeling and Discrete Event Simulation*, Academic Press, New York.