Chapter **4**

Systems Modeling and Analysis

PAUL D. WEST, Ph.D.
JOHN E. KOBZA, Ph.D.
SIMON R. GOERGER, Ph.D.

All models are wrong, some are useful.

—George Box

All but war is simulation.

—U.S. Army

4.1 INTRODUCTION

Thinking about a system's essential components, attributes, and relationships in abstract ways occurs in various forms throughout the system life cycle. This was described in Section 2.8 during the discussion on the spatial placement of systems and their components. It may also take the form of a mathematical representation, such as $E = mc^2$, that examines the relationship between available energy, an object's mass, and the speed of light. At times it is useful to create a physical mockup of an object and put it in the hands of a user to obtain feedback on how the system performs or reacts to change. All of these ways of thinking about systems are using *models* to better understand the system. This chapter explores models—what they are and how they're used—and a unique type of model, a *simulation*, which exercises a system's components, attributes, and relationships over time.

Decision Making in Systems Engineering and Management, Second Edition
Edited by Gregory S. Parnell, Patrick J. Driscoll, Dale L. Henderson

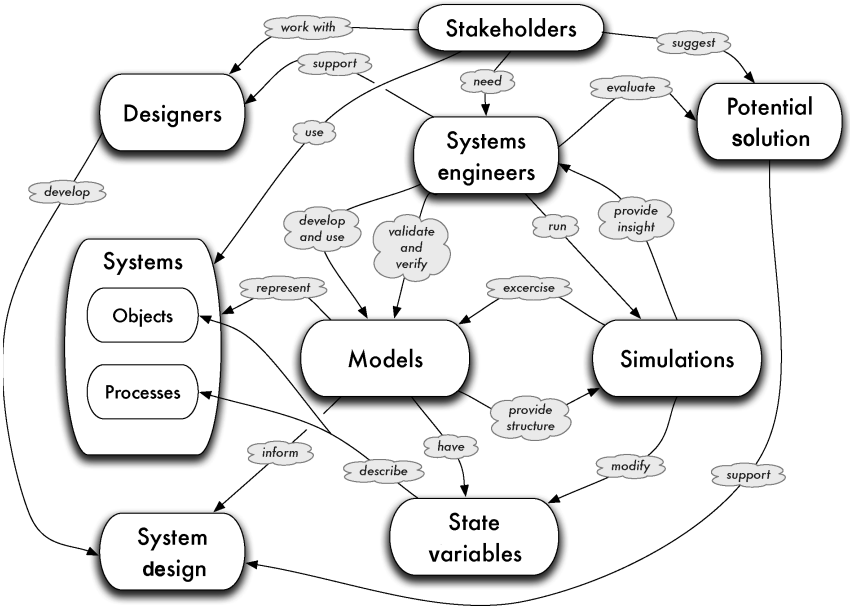

Figure 4.1 Concept map for Chapter 4.

This chapter introduces four of the tools a systems engineer needs to master to fully understand the system life cycle, introduced in Chapter 3, and to develop and evaluate candidate solutions to present to a decision maker. These tools are: *measures of system performance and effectiveness*, *models*, *simulations*, and *determining sample size*. Each tool builds on the previous one. Figure 4.1 captures the fundamental relationships between systems engineers and the modeling and simulation (M&S) process.

4.2 DEVELOPING SYSTEM MEASURES

System measures enable systems engineering teams to evaluate alternatives. They are vital for deciding which models or simulations to use and are therefore key considerations throughout the M&S process. It is essential that measures *based on stakeholder values* be identified before any modeling and simulation effort begins. A fatal flaw in system analysis occurs when modelers force a system into favorite analysis tools, observe dominant outcomes, and attempt to retrofit measures to outcomes. Almost always these measures do not reflect the values of the stakeholders or even the modeler and thus fail to provide meaningful support to the decision maker.

Measures are tied either directly or indirectly to every system objective identified in the problem definition process. Although most objectives can be evaluated directly, others, such as "To develop a sound energy policy," can be assessed

only in terms of percentage of fulfillment of other objectives [1]. Measures can take several forms, but the most prevalent fall into the categories of *measures of effectiveness* (MOE) and *measures of performance* (MOP).

> A *measure of performance* is a quantitative expression of how well the operation of a system meets its design specification.
>
> A *measure of effectiveness* is a quantitative expression of how well the operation of a system contributes to the success of the greater system.

For example, the measure *miles per gallon* reflects the reliability of how well an engine performs. It is not directly related to the success of the greater system's mission, which may be to transport people across town. An effectiveness measure for such a vehicle may better be stated in terms of *passengers delivered per hour*.

A military weapon system may meet a specification of firing 1000 *rounds per minute* (an MOP), but if the *probability of hitting* a target with a single round (an MOE) is 0.9 and the *probability of destroying the target, given a hit* (an MOE), is 0.9, then the measure of performance, rounds per minute, is not very helpful in evaluating the effectiveness of the greater system in its mission of destroying targets.

Performance measures are developed by systems engineers during the Problem Definition phase of the SDP and are described in the system specification. Effectiveness measures are also based directly on stakeholder values, but these measures view the system in a larger context and therefore selecting meaningful MOE can be more difficult. In the 1970s, the U.S. Army produced a seven-step format for defining MOE that is helpful both in the Problem Definition phase of the SDP and for describing the relevance of a measure to stakeholders. In its discussion of MOE, the Army noted several characteristics of a good MOE [2]. These are:

- A good MOE reflects and measures functional objectives of the system.
- A good MOE is simple and quantifiable.
- A good MOE measures effectiveness at echelons above the system (how it contributes).
- A good MOE involves aggregation of data.
- A good MOE can be used to determine synergistic effects of a system.

Defining a good MOE has seven steps, two of which are optional. These are:

1. Define the measure. Include both a narrative description and an equation.
2. Indicate the dimension of the measure. Is it a ratio, an index, a time interval?
3. Define the limits on the range of the measure. Specify upper and lower bounds.
4. Explain the rationale for the measure. Why is this measure useful?
5. Describe the decisional relevance of the measure. How will this measure help the decision maker?

Time To Estimate Range

1. *Definition of the Measure.* Time to estimate range is the elapsed time from detection of a target to estimation of range. Input data are the moment of detection and the moment estimation of range is complete. Relation of output to input is:

   time to estimate range = time of estimation − time of detection.

2. *Dimension of the Measure.* Interval-elapsed time in terms of seconds. If the measure is taken at different times or under varying circumstances, it can be used in the form of mean time to estimate range or median time.
3. *Limits on the Range of the Measure.* The output can be zero or any positive value. The resolution of the measure is limited by the precision of taking start time and end time. The data cannot be disassociated from the definition of computed estimation used, whether it is the first estimate stated regardless of accuracy or is the final in a series of estimates which is used for firing.
4. *Rationale for the Measure.* This measure addresses a component of target acquisition time. Problems in estimation are assumed to contribute to the length of estimation time.
5. *Decisional Relevance of the Measure.* This measure can be used to compare estimation times of means of range estimation (techniques, aids, range finders, trained personnel) with each other or a standard. It would not ordinarily be used alone, but would be combined with accuracy of estimation or accuracy of firing in most cases.
6. *Associated Measures.*
   - Accuracy of range estimation
   - Firing accuracy
   - Time to detect
   - Exposure time
   - Time to identify

Figure 4.2 A fully defined measure of effectiveness (MOE).

6. List associated measures, if any, to put this measure in context (optional).
7. List references (optional). Give examples of where it has been used.

An example MOE using this format is shown in Figure 4.2.

Thinking about how a system will be evaluated occurs continuously as the understanding of the system evolves. This brief introduction on measures will be expanded in depth in Chapter 8.

4.3 MODELING THE SYSTEM DESIGN

This section introduces the concept of modeling. *Modeling* is more of an art than a science. Just as with playing a musical instrument or painting a picture, some

people have more talent for modeling than others. However, everyone can improve their skill with knowledge of their discipline and practice building models. This section provides insight into what makes good models and presents guidelines for building them.

4.3.1 What Models Are

A *model* is an abstract representation of a system. Consider a model airplane. It could be a 1/72 scale plastic model that comes in a kit. The assembled model has some of the features of the real aircraft, such as the shape of the fuselage and the angle of the wings. However, it does not have a working engine or controllable flight surfaces.

Another airplane model is the ready-built radio controlled (R/C) airplane. It flies with the user controlling the elevator, ailerons, and speed. Electric ducted fans even replace jet engines! However, this model is also not the real aircraft. The weight and balance are different, the rudder is fixed, and the flight dynamics are different than the real aircraft.

Both of these are useful representations of the real aircraft system, but each has a different purpose. The model kit illustrates the shape of the fuselage in relation to wing angle. The R/C model demonstrates how the aircraft would look in the sky. Neither model is useful for predicting the maximum altitude or flying range of the actual aircraft. Their usefulness, as in all models, depends on what aspect of the actual system is being studied.

A model captures essential attributes and relationships between system components, but it does not capture them all.

> A model is an abstract representation of a system.

The *modeler* must choose what to put in and what to leave out of a model, as well as how to represent what is put in. This is the essence of modeling. It is also what makes modeling an art. It can be very difficult to choose the key aspects of the system and incorporate them into a workable model. A set of differential equations may accurately represent the relationships among system variables, but it is not workable if it cannot be solved. Modelers build models using their knowledge of the system (or experience with similar systems) and their understanding of available modeling techniques.

4.3.2 Why We Use Models

One of the greatest ideas the Wright brothers had was to model the wing instead of building and flying different wing shapes. In the following excerpt, Wilbur Wright describes how they developed their understanding of the wing. It illustrates many of the reasons to use a model.

> It took us about a month of experimenting with the wind tunnel [Figure 4.3] we had built to learn how to use it effectively. Eventually we learned how to operate it so

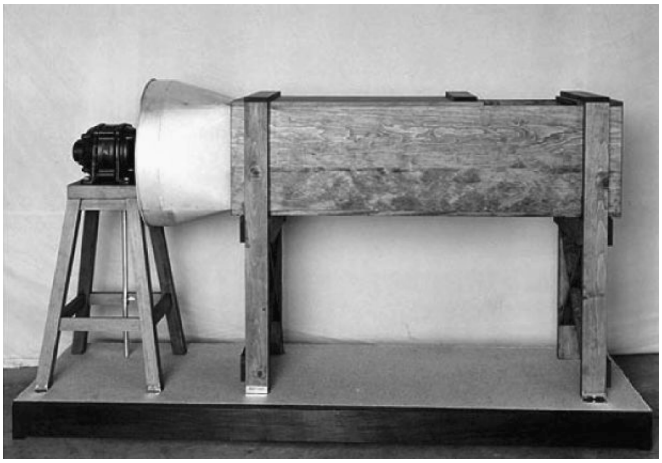

Figure 4.3 Replica of Wright brothers' wind tunnel, 5-10-39 [3].

that it gave us results that varied less than one-tenth of a degree. Occasionally I had to yell at my brother to keep him from moving even just a little in the room because it would disturb the air flow and destroy the accuracy of the test.

Over a two-month period we tested more than two hundred models of different types of wings. All of the models were three to nine inches long. Altogether we measured monoplane wing designs (airplanes with one wing), biplanes, triplanes and even an aircraft design with one wing behind the other like Professor Langley proposed. Professor Langley was the director of the Smithsonian Museum at the time and also trying to invent the first airplane. On each little aircraft wing design we tested we located the center of pressure and made measurements for lift and drift. We also measured the lift produced by wings of different aspect ratios. An aspect ratio is the ratio or comparison of how long a wing is left to right (the wing span) compared to the length from the front to the back of the wing (the wing chord). Sometimes we got results that were just hard to believe, especially when compared to the earlier aerodynamic lift numbers supplied by the German Lillienthal. His numbers were being used by most of the early aviation inventors and they proved to be full of errors. Lillienthal didn't use a wind tunnel like Orville and I did to obtain and test our data.

We finally stopped our wind tunnel experiments just before Christmas, 1901. We really concluded them rather reluctantly because we had a bicycle business to run and a lot of work to do for that as well. It is difficult to underestimate the value of that very laborious work we did over that homemade wind tunnel. It was, in fact, the first wind tunnel in which small models of wings were tested and their lifting properties accurately noted. From all the data that Orville and I accumulated into tables, an accurate and reliable wing could finally be built. Even modern wind tunnel data with the most sophisticated equipment varies comparatively little from what we first discovered. In fact, the accurate wind tunnel data we developed was so important, it is doubtful if anyone would have ever developed a flyable wing without first developing

these data. Sometimes the nonglamorous lab work is absolutely crucial to the success of a project [3].

A couple of bicycle mechanics from Ohio used models to become the first to achieve controlled powered flight. Their use of models saved them time, money, and, probably, their lives. Otto Lillienthal's data was gathered from over 2000 glider tests over more than 20 years [4]. He was killed in a glider crash in 1896. After three months of studying physical models in a wind tunnel, the Wright brothers understood how wings behave. This understanding allowed them to create a flyable wing design for their plane.

This example illustrates the three major reasons we use models.

Models are Flexible. By changing parameter values, a single model can represent the system across a broad set of conditions or represent a variety of related systems.

Models Save Time. It is usually much faster to build or change a model of a system than the system itself. Testing the design using a model could reveal flaws that can be fixed more quickly and less expensively before the system is built.

Models Save Money. Building prototypes of a car or a production line is expensive. It is much cheaper to make changes to a mathematical or computer model than to build different versions of the actual system.

4.3.3 Role of Models in Solution Design

The primary purpose of a model is to understand how the actual system design will or does perform. This understanding can then be used to design the system in a way that improves or optimizes its performance. The Wright brothers used their wind tunnel to gain an understanding of how a wing behaves. They then used this knowledge to design a wing for their aircraft.

There are three different types of system designs: *satisficing, adaptivising*, and *optimizing* (Figure 4.4). *Satisficing* is a term that comes from a concatenation of *satisfying* with *sacrificing*. Satisficing, as a goal for an existing system, means the current performance is satisfactory. For a new system, satisficing means that any good feasible solution with acceptable intrinsic cost tradeoff will be satisfactory. If the goal is to move up a hillside, a satisficing solution may be to crawl uphill through some bushes.

The primary criterion for *adaptivising* solutions is cost effectiveness. Improved system performance is the goal, but only when it can be obtained at a reasonable cost. Satisficing accepts any solution that works. Adaptivising looks for "good" solutions. For the example of moving up the hill, an adaptivising solution may be to walk a path that leads up the hill. Ascent may not be as rapid as crawling up a steep slope, but it is a lot easier and will probably get us there faster.

Optimizing solutions are at least as good and often better than all other solutions. They are the best according to performance measures. However, they may require

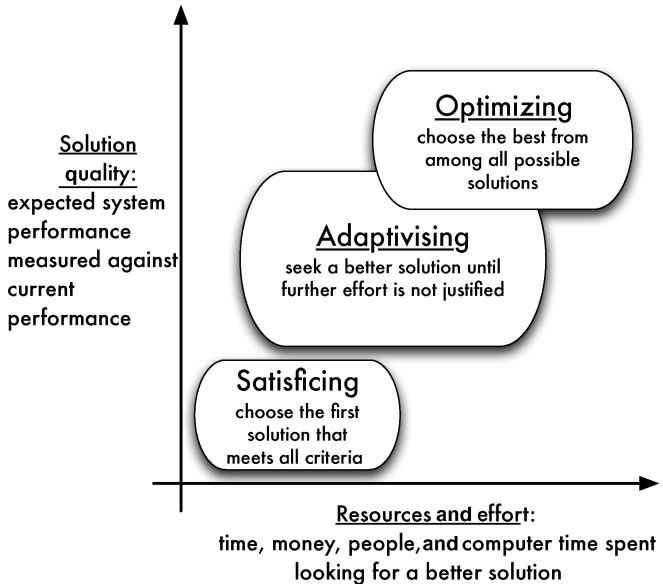

Figure 4.4 The relationship between effort and solution quality.

much effort or expense. If the goal is to get to the top of the hill in the quickest manner, the optimizing solution may be to land on top of the hill in a helicopter.

4.3.4 Qualities of Useful Models

> Art is a selective recreation of reality. Its purpose is to concretize an abstraction to bring an idea or emotion within the grasp of the observer. It is a selective recreation, with the selection process depending on the value judgments of the creator [5].

The above definition of art sounds perilously close to our definition of a model. *Aesthetics* is the branch of philosophy that relates to art and includes the study of methods of evaluating art. Just as art can be evaluated, so can models. Just as the artist must make value judgments when capturing an idea or emotion in a concrete form, so the engineer must make value judgments when representing the key elements of a system in a usable form. Just as art can be beautiful, so can models. This section presents the characteristics used to judge models.

Parsimony. One of the primary characteristics of a model is its level of complexity. The principle of parsimony (also known as Occam's Razor) states that when choosing among scientific theories that make exactly the same predictions, the best one is the simplest [6]. The same is true of models—simpler is usually better. Given several models of the system with comparable output, the best model is the simplest one. It is typically easier to understand. It also

may require fewer and more reasonable assumptions. For these reasons, it is usually easier to explain to stakeholders.

Simplicity. Another characteristic is expressed by the relationship between the complexity of the model and that of the system it represents. The model's level of complexity typically increases in response to the system's level of complexity. This is not always the case, but a complex model for a simple system usually represents a lack of effort on the part of the modeler. It takes time, effort, and creativity to simplify a model. Sometimes this can result in a simple model of a complex system, which is truly a thing of rarest beauty.

Accuracy. Accuracy is another important characteristic of models. Accuracy is "the degree to which a parameter or variable, or a set of parameters or variables, within a model or simulation conforms exactly to reality or to some chosen standard or referent" [7]. An inaccurate model is not much good to anyone. Accuracy must often be considered hand-in-hand with model complexity. A "quick-and-dirty" solution results from a simple model with low accuracy. Sometimes this level of accuracy is enough to gain sufficient insight into the system behavior. At other times, more accurate, and perhaps more complex, models are needed.

Robustness. Robustness characterizes the ability of a model to represent the system over a wide range of input values. A model that is not robust only represents the system for a narrow range of inputs. Beyond this range, other models must be used or the structural parameters of the existing model must substantially change. This is not necessarily a problem. For example, if the narrow range of input values covers all of the alternatives from the Solution Design phase of the SDP, the model is good. However, if the narrow range only covers a portion of the design space, other models will be needed; or a more robust model must be developed.

Scale. The scale of the model is an important characteristic that refers to the level of detail used in the problem. If a model contains both insect life cycles and continental drift, it may suffer from inconsistent scale. Both ends of the spectrum may exist in the actual system, but there are orders of magnitude difference in how they affect the system. If you are modeling the life cycle of a flower, the effects of continental drift are negligible. However, if you are modeling the spread of a flower species over geologic periods, continental drift is much more important than insect life cycles.

Fidelity. Fidelity is an overall characterization of how well the model represents the actual system. Just as the high-fidelity recordings of the late vinyl age were touted as being truer to the actual sound produced by an orchestra, a high-fidelity model is closer to representing the actual system. Fidelity is an aggregate characteristic that brings together complexity (parsimony and simplicity), accuracy, robustness, and scale. A 100% replication of an actual system represents a duplicate or copy of the actual system.

Balance. Just as an artist must choose which aspects of an idea to convey and how to represent it in an art form, a systems engineer must choose which

aspects of a system are essential and how to represent them in the concrete form of a model. The modeler must balance and blend complexity (parsimony and simplicity), accuracy, scale, and robustness. (One definition of an expert is knowing what to ignore.) Among the choices the modeler must make is the type of model (or models) to use. Section 4.5 introduces some of the different types of models.

4.4 THE MODELING PROCESS: HOW WE BUILD MODELS

Just as there is no "correct" way to create art, there is no 'correct' way to create a model. However, there are steps or processes that have been used in the past to develop models. This section presents only one possible modeling process. Think of it is a list of things to consider while developing a model, rather than a rigid step-by-step procedure. In fact, some of the steps listed in Figure 4.5 may need to be repeated several times in different orders as the model is developed.

Typically, a model begins in the mind and is then rendered in a form that is tested or examined for usefulness. The experience gained may then be used to revise and improve the model. This sequence is repeated until an acceptable model is created, verified, validated, and accredited.

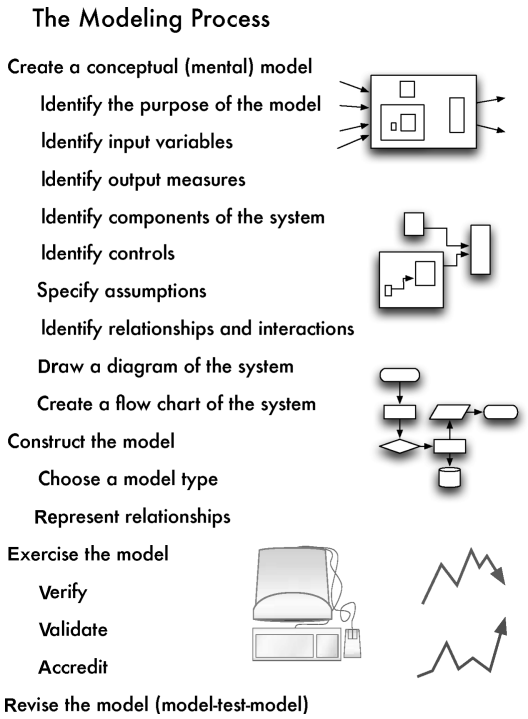

Figure 4.5 The modeling process.

4.4.1 Create a Conceptual Model

Not surprisingly, the *conceptual model* begins in the modeler's mind as the realization of a key relationship or an idea about representing a key feature of the system. For example, the modeler may notice that inventory goes up as demand decreases or production increases. How can this relationship be represented in a model?

> A *conceptual model* is a theoretical representation of the system based on anecdotal or scientific observations. "It includes logic and algorithms and explicitly recognizes assumptions and limitations" [8].

Many of the items or steps in Figure 4.5 related to the conceptual model have already been considered as part of the systems decision process (see Chapter 2). These include identifying system elements or components, specifying assumptions, analyzing data to identify relationships and interactions, and drawing diagrams and flowcharts for the system. Other items need to be considered in the context of the desired model.

Identify the Purpose of the Model. The most important consideration is the purpose of the model. One purpose of a model might be to predict how a system will perform under a given set of conditions. Consider the following examples: How many satellites can be placed into space if two space shuttles or payload rockets are purchased? What is the worst-case response time to a distress call from manned lunar station? What penetration will a specific rocket thrust or certain projectile shape achieve through a concrete block? Another possible purpose could be to determine the set of design choices that optimizes performance as in the following examples. How many machines should be purchased to maximize profit? What balance of forces should be used to maximize the probability of success? What is the best fin configuration for a rocket system? The purpose of the model will influence the choice of modeling tools used.

Identify Input Variables. The purpose of the model affects the choice of model *input variables*. These could represent aspects of the design that the systems engineer can choose—sometimes called controllable or independent variables. These could include such factors as the number of space shuttles, payload delivery rockets, or emergency response teams. They can also represent parameters that the user of the model may be unsure of or wishes to vary across a range of values as they test the model exploring "what if?" scenarios. These may include the rate of inflation, the cost of rocket fuel, rocket fin configuration, the amount of thrust produced by a rocket engine, or the thickness of the asbestos shielding on the space shuttle.

Identify Output Variables. Output variables represent what the modeler wants to learn from the model. These are sometimes called uncontrollable or dependent variables, and may include the amount of payload that can be taken into space, the response time for an emergency response team or the number of rockets to purchase.

Identify Controls. Identifying inputs, outputs, controls, and components (mechanisms) of a system provide the elements necessary to build in IDEF0 model. Of those, controls are often the most difficult to understand. The IEEE Standard for IDEF0 (IEEE Std 1320.1-1998) defines a control as "a condition or set of conditions required for a function to produce correct output." This may include a wide variety of conditions, including regulations, instruction manuals, or "trigger" events. Section 2.6 lists the family of IDEF models and points out that IDEF0 focuses on functions. IDEF0 models must have at least one control and at least one output, although typically a function will also include at least one input, which is transformed into an output by the function.

Interestingly, the same variable could be an input variable for one model and an output variable for another. For example, suppose a model is created to predict the probability of detecting an explosive device in a bag that is screened by Machine A and Machine B. An input value could be the detection rate for Machine B at an airport terminal. However, if the purpose of the model is to determine the detection rate for Machine B that maximizes the probability of detecting the device, then the detection rate for Machine B is an output variable.

After the modeler understands the system and has created a conceptual model, the model can be built.

4.4.2 Construct the Model

One of the key decisions in constructing a model is choosing a type of model to use. Different types of models and some of their implications will be discussed in Section 4.5. For now, we will focus on some of the different factors that lead to the choice of a model type.

> A *constructed model* is an implementation of a conceptual model—mathematical, physical, or codified.

Choose a Model Type. Sometimes assumptions about the system lead naturally to a choice of model. If it is assumed that there is no randomness in the system, then deterministic models are appropriate (Section 4.5). The availability of data can also drive the choice of a model type. For example, lack of data for a certain input variable may lead to another model type that does not need that input. The type of data is also a factor. For example, quarterly data may lead to a discrete-time model that moves in 3-month steps. In addition, the goal of the model can be the primary factor in choosing a model type. For example, optimization models are likely to be used if the purpose is to find the best solution.

Represent Relationships. After the model type is chosen, the system relationships relevant to the goal must be represented in the model. Each type of model has specific ways of representing relationships. For example, a linear

model $y = mx + b$ is defined by the choice of m and b. A continuous-time Markov chain is described by the definition of the state variables and the transition rates.

Issues may arise during the M&S process which require rethinking the conceptual model. For example, perhaps there is insufficient data to determine the value of m in a linear model or perhaps the range of data is too narrow relative to the desired range of system operation.

4.4.3 Exercise the Model

Once a model is constructed, the modeler must determine how good the model is and how and when it is appropriate to use. This is referred to as exercising or testing the model. There are several concepts related to this testing: verification, validation, and accreditation.

Verification. Verification determines whether the constructed model matches the conceptual model [9]. That is, was the conceptual model correctly implemented? First, simple tests are done—sometimes referred to as sanity checks. Are there special cases where the behavior or condition of the conceptual model is known? For example, in the conceptual model if there is a limited payload and no rocket launch, there should be no satisfied demand for resupplies needed at a manned lunar base. Does this happen in the constructed model?

If the model passes the simple tests, more complicated ones are tried until the model developer and accreditation agent are confident that the constructed model performs as the conceptual model states it should. Input variables and parameters may be varied over a range of values to determine the model's accuracy and robustness. If a conceptual model predicts that inventory will rise as demand decreases, does this occur in the constructed model? Actual data from the system or a similar system may be used as input to the model, and the output can then be compared to the actual behavior of the system.

During this process, failure can be more enlightening than success. Parameters may need to be adjusted or extra terms or components added to the model that were initially overlooked because of tacit assumptions that were not intended. Sometimes the model does not represent the system well, and a new type of model must be developed.

Validation. Validation determines whether the conceptual model (and constructed model if it is verified) appropriately represents the actual system [9]. For example, a conceptual model may assume a linear relationship between thrust and speed; that is, speed increases proportionately as thrust increases. If the actual system includes both drag and thrust, then there may be times when speed declines because drag from the payload and atmospheric conditions are greater than the force generated by thrust. In this case, the conceptual model is not valid.

Validation is concerned with building the right model. Verification is concerned with building the model right. Both concepts are important and both need to be checked when testing a model. Testing is usually not separated for verification or validation. Tests are conducted and failures are investigated to determine if the problem is with the conceptual or constructed model. Previous steps in the process may need to be repeated to correct the problems.

> Validation is building the right model. Verification is building the model right.

Accreditation. Accreditation is an "official" seal of approval granted by a designated authority [9]. This authority is frequently the decision maker, but may be an accreditation agency, such as ABET, Inc., which accredits academic engineering programs. This signifies the model has been verified and validated for an intended purpose, application, or scenario. Accreditation represents the process of users accepting the model for use in a specific study. This can be very informal, for example, if the same people who develop the model will be using it. On the other hand, if the model is developed by one organization and used by another, accreditation can be a lengthy and involved process. Normally, models are accredited by a representative of the using organization. All models and simulations used by the Department of Defense must be verified, validated, and accredited [9].

It may be tempting to believe that a model is only judged by its usefulness. However, the process used to create the model and the testing involved in its development are both very important when judging a model. What characteristics were considered for inclusion in the model? What assumptions were made while developing the model? What value ranges or conditions were tested? The answers to such questions build confidence in the model's capabilities.

4.4.4 Revise the Model

As has already been discussed, test results may cause changes in the model. This is sometimes called the *model–test–model* concept. The idea is to use knowledge gained about the system or the model to continually revise and improve the conceptual or constructed model. Revising the model is always an option and typically happens quite frequently during model development.

Recall the story of the Wright brothers' wind tunnel. Their experience flying gliders and data from others helped them to know what they wanted to measure and what they wanted to vary. Lift and drift were their output variables. The different types of wing shape and aspect ratios were their input variables. The wind tunnel gave them the means to do it. They spent over a month experimenting with the wind tunnel. This corresponds to verification and validation. They wanted to be confident that their model would yield useful results and they understood the conditions under which it would. At some point they had a discussion and agreed they were ready to collect data. This corresponds to accreditation. Once they were confident of how to

use their model, they systematically explored the system by trying different wing shapes and aspect ratios.

4.5 THE MODEL TOOLBOX: TYPES OF MODELS, THEIR CHARACTERISTICS, AND THEIR USES

There are several types of models including physical, graphical, and mathematical. Even though these are distinct types of models, they are often used together as will be discussed below.

Physical. Physical models involve a physical representation of the system, such as the wing designs the Wright brothers tested in their wind tunnel. These models can be time-consuming and expensive to produce. However, they can provide valuable data that can be studied to learn about the system. Sometimes physical models are the only alternative because mathematical models may not be practical. This was the case for many aerospace models until high-speed computers allowed real-time solution of mathematical formulas for fluid flow dynamics to replace, or reduce the number of, physical wind tunnel tests.

Physical models need not always be miniature versions of the real system. Sometimes a physical representation is used to take advantage of a physical property. For example, soap film takes a shape that minimizes surface area. Before the age of computers, some optimization problems were solved using soap films and wire frames or glass plates with wooden pegs. Surprisingly, this technique is currently a topic of discussion concerning a fundamental research question related to computational complexity theory [10].

Sometimes physical models are based on analogies; something in the physical model represents a quantity of interest in the system. An interesting example of this is the analog computer. An analog computer uses electrical circuits to represent differential equations. Combinations of operational amplifiers, resistors, and capacitors are used to construct circuits that perform such mathematical operations as integration, differentiation, summation, and inversion. The Norden bombsight used during World War II was a mechanical analog computer that calculated the trajectory of the bomb and determined when to release it.

Physical models have always had limited areas of application, and the use of digital computers has further limited their use. However, they can still effectively demonstrate system performance. Seeing a truss design made out of toothpicks support a large weight can be pretty dramatic!

Graphical. Graphical or schematic models use diagrams to represent relationships among system components. Examples are causal loop diagrams that represent feedback relationships or Petri networks that represent a sequence of conditions over time. Figure 4.6 shows a network model of a discrete event simulation (see Section 4.6). These can be used to show alternatives.

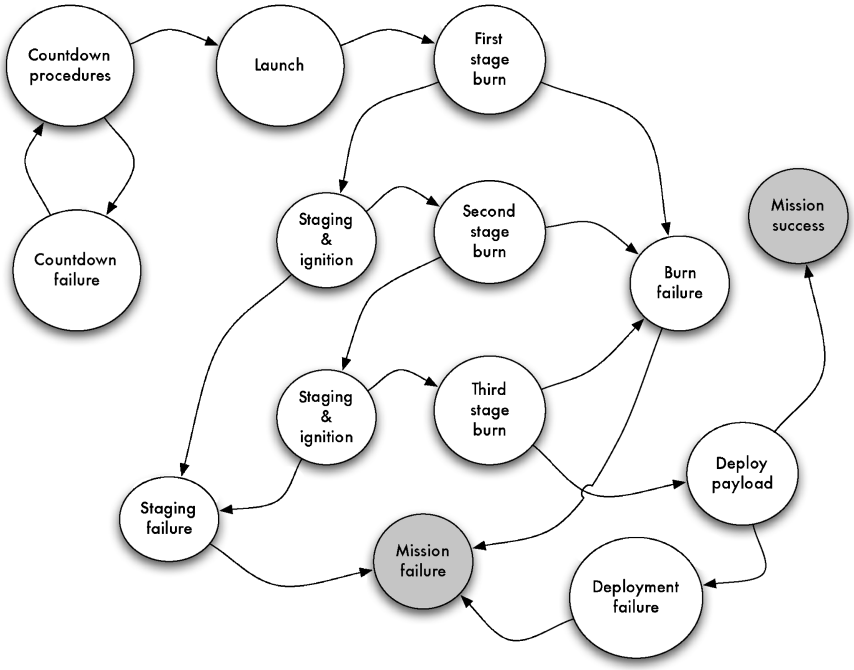

Figure 4.6 Rocket launch discrete event model.

Concept maps, such as those used at the start of each chapter in this book, can represent how different system characteristics or components affect one another. Sometimes these diagrams are the end product of the systems decision process and can provide an understanding of some aspect of the system being studied. Sometimes they are graphical front-ends for computer packages that generate and execute code. The code may represent mathematical equations based on the diagrams, which are numerically solved to provide quantitative results. System dynamics software is an example of such a package. The relationships in the diagram are represented as differential equations that are numerically solved by the computer. In other packages, such as the discrete-event simulation languages Arena® and ProModel®, the graphical model represents the logical structure of simulation model (see Section 4.6). The computer uses this to generate code that is run to produce both numerical and graphical output. For example, the computer may show people or jobs moving through the system.

Mathematical. Mathematical models use quantitative relationships to represent systems. It can be helpful to view mathematics as a language, such as Latin or Spanish. Learning a new language involves vocabulary and syntax. Vocabulary refers to the meaning of words. Syntaxis learning how to put words together to convey ideas. Mathematics also has a vocabulary and syntax. Just as words in a novel convey complex thoughts, emotions, and relationships,

mathematics can convey the complex relationships and structure of a system. Part of the challenge of constructing a mathematical model is translating from English to math.

Mathematical models can provide exact or approximate solutions. Approximations are models that provide a solution close to the actual solution. Although not as accurate as exact solutions, approximations are usually much easier to solve. As an example, suppose we wish to determine the area under the function $f(x) = x^2$ between points a and b. The exact solution is

$$\int_a^b x^2\,dx = \frac{1}{3}x^3\Big|_a^b = \frac{b^3 - a^3}{3}. \tag{4.1}$$

An alternative is to use an approximate model to represent the area with a trapezoid. The approximate solution would be

$$\left(\frac{a^2 + b^2}{2}\right)(b - a) = \frac{b^3 - ab^2 + a^2b + a^3}{2}. \tag{4.2}$$

If $a = 1$ and $b = 2$, the exact solution is 7/3 and the approximate solution is 5/2. The approximation error is 1/6. The advantage of the approximate solution is that it did not require integration.

Mathematical models can be solved analytically or numerically. An analytical solution is a mathematical expression that yields a value after appropriate parameter values are substituted. For an analytical solution, the hard work is usually done before the parameter values are given. They are represented as variables or constants in the final expression. The values are substituted into the expression and some arithmetic is performed to obtain the solution. A numerical solution requires a (sometimes lengthy) series of calculations after the parameter values are given.

Again consider the area under the function x^2 between points a and b. Both of the solutions, Equations (4.1) and (4.2) in the previous paragraph are analytical solutions. Instead of using constants such as a and b, a numerical (or computational) solution needs numbers from the beginning. If $a = 1$ and $b = 2$, a numerical solution approach may divide the interval between 1 and 2 into some number of steps—in this case, 10. For each step, the area under the curve could be approximated as a rectangle (see Figure 4.7). Adding the area of each rectangle gives the solution.

Numerical solutions often use approximations, which can introduce approximation error. They may also have errors due to computation, such as rounding or truncation. For example, consider the expression $3(4/3)$. If it is evaluated analytically, the result is 4. If it is evaluated numerically on a computer, the result is 3.9999999, because the computer cannot represent the exact value of $1/3$.

Computers can be used to obtain either analytical or numerical solutions. Mathematical analysis software such as Mathematica®, Matlab®, and

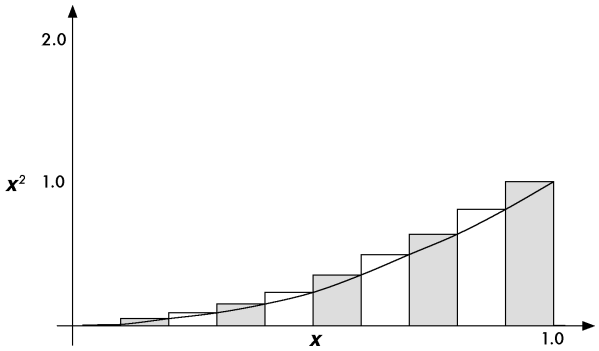

Figure 4.7 Area under the curve.

Maple™ can solve many problems analytically. They also provide a variety of numerical tools to solve problems.

4.5.1 Characteristics of Models

Models have a number of *characteristics* that can be used to determine when they are appropriate to use. Some of these characteristics are shown in Figure 4.8 and are discussed below. These characteristics will be presented as either/or possibilities—a model is either static or dynamic—however, models often have elements of both.

Descriptive versus Prescriptive. Descriptive models describe or predict how a system will behave. Their outputs give information about system behavior. *Prescriptive* models prescribe a course of action—they dictate what the best thing to do. A descriptive model of a payload rocket launch could represent

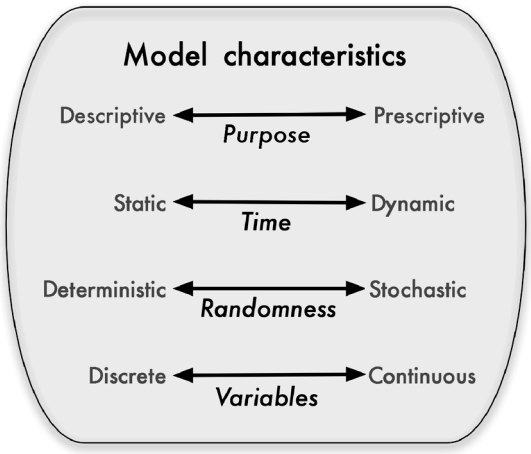

Figure 4.8 Model characteristics.

the pre-launch, launch, and post-launch activities at the launch pad to compare the different possible locations for launch observers, security forces, emergency response teams, and launch monitoring cameras. A prescriptive model would be used to determine the proper mixture of rocket fuel and payload. Consider a scheduling problem in which a number of different jobs must be processed on a machine. There are different types of jobs, and a changeover on the machine is needed when switching from one type of job to another. A descriptive model would be used to determine the total production time for a given sequence of jobs. A prescriptive model would be used to determine the sequence with the shortest total production time.

How many boxes are needed to cushion the fall of a motorcycle and rider jumping over an elephant? Several models are needed to solve this problem. A descriptive model is needed to describe the speed and momentum of the motorcycle and the rider over time. The values of these quantities at the moment of impact (hopefully with the boxes not the elephant) would be an input to a prescriptive model that would determine the number of boxes.

Static versus Dynamic. Static models do not change. *Dynamic* models represent change. This change usually occurs over time, but could also be over distance (e.g., gravitational or magnetic attraction) or some other measure. As an example of a problem that is static, suppose that a company has a development budget and a large number of potential products, each of which has a development cost and a potential profit. Which products should be developed to maximize the potential profit? The parameters of this problem do not change. They are static. Change in the model usually refers to the primary output of the model and whether or not it changes. In this problem, there is only one list of products; it does not change. If the potential profit for each product changed over time to reflect changes in consumer needs or tastes, the primary output of the model, the list of projects to invest in today, is still static. However, if the company had annual budgets for each of the next five years and wanted to know which projects to invest in each year, the output becomes dynamic because the list changes each year.

As another example, suppose there is a rocket manufacturing plant. A dynamic simulation model could represent arriving raw materials (metal alloys, fiber-optic cable, wiring harnesses, computer processors, instrument gages, etc.), storing these materials, modifying them into rocket components, assembling the components into a rocket, and delivering the rockets to the customer. The simulation could monitor inventory levels over time, the time in system for the different materials, or the occurrence of shortages.

Dynamic models can represent transient or equilibrium behavior. *Transient-focused* models concern behavior that depends on the initial conditions of the system. They are usually appropriate over short time intervals. *Equilibrium-focused* models represent system behavior after it is no longer affected by initial conditions. They are usually appropriate over longer time intervals. The amount of fuel left in the launch rocket after the initial launch phase depends

a lot on the amount of fuel and payload in the rocket prior to ignition. This is transient behavior. The amount of fuel in the space shuttle prior to reentry does not depend on how heavy the initial payload was (except in extreme cases) since another rocket subsystem launched the shuttle into space.

Deterministic versus Probabilistic. *Deterministic* models use quantities that are known with certainty. *Probabilistic* (or *stochastic*) models have at least one quantity with random values. Suppose the chairs of a chair lift arrive at the top of a ski slope every 30 seconds. If every other chair has one skier and the rest have two, how many skiers are at the top of the slope after 10 minutes? All the quantity values are known with certainty, so a deterministic model is appropriate here. If the arriving chair has one skier with probability 0.5 or two skiers with probability 0.5, how many skiers are at the top of the slope after 10 minutes? The number of skiers on a particular chair is not known and the number of skiers after 10 minutes cannot be known with certainty. A probabilistic model is appropriate here. A deterministic model of the rocket production plant's inventory would use fixed quantities for the times between material deliveries, production service times, and pickup times. The model would still be dynamic and inventories would change over time, but there would be no variation and, eventually, a repeating pattern will appear. A probabilistic model would introduce random variation into at least one of the quantities, which would result in continually changing behavior over time.

Discrete versus Continuous. The variables in a problem can be discrete, continuous, or mixed. One way to tell is to put all the possible values for the variable in a set. If the number of values is countable (i.e., a one-to-one correspondence exists with the positive integers), the variable is discrete. *Discrete* means separate, distinct, or unconnected. In this case, the values are separate so we can distinguish and count them. Obvious examples are the number of rocket fins in the rocket assembly plant inventory at noon each day or the number of cars passing through an intersection on a given day. Another example of a discrete problem we have already discussed is the set of projects to fund. Different sets of projects are distinct from each other and we can count how many different possibilities there are. The sequence of jobs to process is another example.

Continuous variables represent quantities that take on values from an interval, such as $0 \leq x \leq 10$ or $-501.6 \leq y$. How many values are between 0 and 1? How do you separate them? Is 0.5 different than 0.50000001? An example of a continuous quantity is the total production time for a sequence of jobs or the average time it takes a rocket to enter near space once the launching rockets have ignited. Mixed variables take on values from both countable and uncountable sets.

4.5.2 The Model Toolbox

Several different tools can be used to fix one board onto another. A nail gun uses compressed air to drive the nail and is ideal for applications requiring many nails.

However, it may be too expensive or take too much time to set up to drive a single nail. A claw hammer is relatively inexpensive, but may require some expertise to use without bending the nail or damaging the wood when missing the nail. If neither of these quality defects is a concern, a heavy wrench or even a rock could be used to drive the nail. All of these tools are options for accomplishing the objective. Each has its advantages and disadvantages, which must be considered when choosing which one to use. The same is true of modeling tools.

This subsection surveys the modeler's toolbox and presents some of the options available when deciding which type of model to use. To aid in understanding how these tools relate to each other, the toolbox is divided into 16 compartments based on the characteristics previously presented (see Table 4.1). Organizing the toolbox this way should help you determine the appropriate tool to use. Similar to the Myers–Briggs Type Indicator, the "personality" of each compartment will be represented by a four-letter code. The sequence represents Static (S) or Dynamic (D), Deterministic (D) or Probabilistic (P), Descriptive (D) or Prescriptive (P), Discrete (D) or Continuous (C).

Ideally, this section would provide a thorough presentation of the tools in each compartment of the toolbox and how to use them. However, since entire courses are dedicated to accomplishing this task for many of these tools, we will, instead, give examples of the problems associated with each compartment, list some of the appropriate tools and suggest helpful references. References that survey many of the compartments include references 11–13. An online source for all things related to operations research and management science is reference 14. Two good references for using spreadsheets are references 11 and 15.

The majority of the following discussion concerns example problems containing characteristics associated with each of the compartments. To further explain each of the compartments, consider the following problem setting. A news vendor sells newspapers on a street corner. The newspapers are purchased from the publisher at \$0.50 and sold for \$0.75. This is called the news vendor problem. We will look at different versions of the problem as we explore each of the compartments.

The first two compartments describe problem characteristics that should be familiar to you. Arithmetic, algebra, and geometry are key tools used throughout the course of secondary and college education. Spreadsheets have helpful built-in functions that automate many of the associated mathematical tasks and are frequently used for large models [11, 15].

TABLE 4.1 The Modeler's Toolbox

Models		Static		Dynamic	
		Deterministic	Probabilistic	Deterministic	Probabilistic
Descriptive	Discrete	1-SDDD	5-SPDD	9-DDDD	13-DPDD
	Continuous	2-SDDC	6-SPDC	10-DDDC	14-DPDC
Prescriptive	Discrete	3-SDPD	7-SPPD	11-DDPD	15-DPPD
	Continuous	4-SDPC	8-SPPC	12-DDPC	16-DPPC

Compartment 1. SDDD (Static, Deterministic, Descriptive, Discrete). For the news vendor problem, what will the profit be if the news vendor purchases 50 newspapers and sells 35 of them? This problem is static; the context is a specific point in time. The data are discrete since the purchases and sales are in whole newspapers. The data are also deterministic because all the quantities are known and fixed. Finally, the problem is descriptive. No decision is needed.

Recall the company that has a development budget and a large number of potential products, each of which has a development cost and a potential profit. Given a subset of the products, compute the total development cost and potential profit for the subset. A spreadsheet model could be used.

Compartment 2. SDDC (Static, Deterministic, Descriptive, Continuous). Some simple examples include computing the number of gallons of water needed to fill a pool that is 15 feet long, 10 feet wide, and 4 feet deep or computing the monthly payment for a $1000 loan at 10% interest compounded annually and amortized over 10 years. The formulas $V = IR$ and $F = ma$ are other examples from this compartment. Suppose the time it takes until the next newspaper sells is 1.5 times the time it took the last newspaper to sell. If it takes 0.1 hour to sell the first newspaper, then what is the total time needed to sell 10 papers? The quantity of interest is now continuous rather than discrete.

Compartment 3. SDPD (Static, Deterministic, Prescriptive, Discrete). As an example of a problem associated with this compartment, let us consider the company that has a development budget and a large number of potential products, each of which has an estimated development cost and a potential profit. Which products should be chosen to maximize the potential profit? A prescriptive solution is needed to identify the best combination of products. Prescriptive problems are usually complicated by constraints. Here, the development budget limits or constrains the possible combinations of products.

Suppose the news vendor also has the option of selling the magazine *Systems Engineering Today*, which can be purchased for $0.75 and sold for $1.50. Suppose up to 20 customers will purchase only a newspaper, 15 customers will purchase both a newspaper and magazine, and five customers will purchase only a magazine. If the news vendor has $15.00 to purchase supplies, how many newspapers and magazines should she purchase to maximize her profit? The problem is now prescriptive instead of descriptive.

A similar problem structure can be seen in a crew scheduling problem. An airline wishes to schedule its pilots and flight attendants for the next month. There are many conditions that the schedule must satisfy. There are union requirements for the number of hours a crew member can be on duty per day or away from their home station. Each crew member's next flight must leave from the destination of their last flight (or they need to be flown as a passenger to their next starting point). The crew members submit work requests, and their seniority must be considered when constructing the schedule. Costs increase if crews stay overnight away from their home stations or fly too

many hours in a day. Of course, it is more costly to have flights without an available crew! The airline wants a schedule that is feasible (i.e., satisfies all the conditions) and has the lowest cost (or at least lower than most of the other schedules).

This is a complicated problem, why does it belong in this compartment? First, it is discrete. A solution is a schedule assigning crew members to flights. Each possible schedule could be numbered and counted. The number of solutions is countable and, therefore, discrete. The airline wants the schedule with the lowest cost. Finding a better solution or the best solution requires a prescriptive model. There is no randomness in the problem, so it is deterministic. Finally, the problem is static; the solution does not change over time. The schedule is only for one month.

Small problems in this compartment can be solved by enumerating all the possible solutions, using a descriptive model to evaluate each one, then choosing the best one. Large problems of this type require discrete optimization tools. It is often difficult to find the best possible solutions for very large problems, so heuristics are often used. These are algorithms that find solutions that are good, but not necessarily the best. Some discrete optimization topics include combinatorial optimization and integer programming. References for tools in this area include references 12 and 13. Note that many prescriptive tools include the term programming (e.g., integer programming or linear programming). In this context, programming means planning. A concert program (or play list depending on the type of concert) is a plan for the music to be performed.

Compartment 4. SDPC (Static, Deterministic, Prescriptive, Continuous). Interestingly, continuous problems are often easier to optimize than discrete ones. Suppose a quantity of interest is described by the formula $y = (x - 3)2$. What value of x gives the minimum value of y? Calculus can solve this problem analytically. Numerical techniques include line-search algorithms such as the bisection search or the golden mean search.

There are many different types of problems in this compartment. The function to be optimized may be linear or nonlinear. There may or may not be constraints. If there are, they may be linear, nonlinear, or both. As a result, many different tools have been developed to model and solve these types of problems.

Linear programming is a tool that can very efficiently solve some problems in this compartment. The function to be optimized and the constraints to be satisfied must all be linear functions. The simplex method [16–18] is an efficient algorithm to find optimal solutions for linear programs and is now widely available in spreadsheet packages such as Microsoft® Excel.

As an example, consider the following product mix problem. A gravel company has two types of aggregate mixes it sells. Each consists of a different ratio of two component aggregates and has different profit margins. The quantities of component aggregates are limited. If the company can sell all

it produces, how much of each mix should be produced to maximize profit? This is the continuous version of the news vendor problem with newspapers and magazines. Instead of whole numbers of newspapers and magazines, the quantities can be in any amount. The problem is still static and deterministic. It is also prescriptive since the best solution is needed—one that will maximize profits.

A primary difficulty faced with most nonlinear problems is distinguishing between *local optimal* solutions and *global optimal* solutions. Figure 4.9 shows a function of x that has several points where the first derivative is zero and the second derivative is positive. These points are local optimal solutions (for a minimization problem); they are better than the nearby points that surround them. A global optimal solution is a local optimal solution with the best value. There are a number of techniques to identify local optimal solutions [19, 20]. *Metaheuristic techniques* [21] such as tabu search, simulated annealing, genetic algorithms, and generalized hill climbing try to move past the local optimal solutions to find the global optimal solutions.

These metaheuristic algorithms are flexible prescriptive tools and are appropriate (with varying degrees of effectiveness) for any compartment with a prescriptive element.

Compartments 5 through 8 are similar to Compartments 1 through 4, respectively, except one or more of the input variables or parameter values are now uncertain quantities. They are random variables with a probability mass function (if they are discrete) or a probability density function (if they are continuous). This randomness could be the result of measurement error or the inability to know with certainty what will happen. One approximation technique is to model and solve the problem as a deterministic one by fixing each random variable at its expected value. This approach can be misleading because the result is typically not the expected value of the solution.

Analytical techniques may be available for these compartments depending on the type of distributions and relationships involved (e.g., sums of normal random variables) [22, 23]. If the discrete random variables have only a few

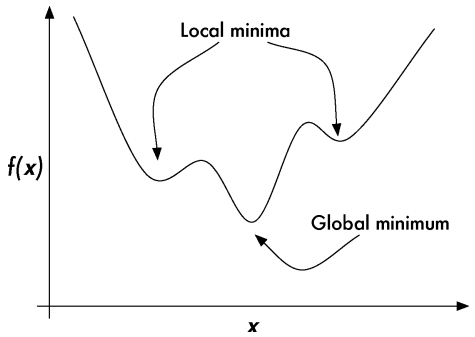

Figure 4.9 Local and global minima of $f(x)$.

possible values, probability tree diagrams can be a useful tool [16]. Statistical tools such as hypothesis testing or regression analysis can be helpful [24, 25]. An easy-to-apply numerical technique is Monte Carlo simulation [11]. Suppose the distribution functions of two random variables X and Y are known and the distribution of $Z = XY$ is needed. The functional relationship is specified in a program or a spreadsheet. Random number generators produce values for X and Y, which are substituted into the function. The resulting value of Z is saved. The process is repeated, perhaps millions of times. The resulting data represent the distribution of Z. Spreadsheet add-ins for this technique include @Risk® and Crystal Ball®.

Compartment 5. SPDD (Static, Probabilistic, Descriptive, Discrete).

Compartment 6. SPDC (Static, Probabilistic, Descriptive, Continuous). The previous versions of the news vendor problem were deterministic and did not consider the fact that the number of newspapers that can be sold on a particular day is not known when the vendor buys her newspapers. Demand is uncertain and dependent on factors such as the weather, local traffic conditions, and world events. The vendor could make more profitable decisions about the number of papers to order if uncertainty were reduced. It may be possible to use a probability distribution to quantify this uncertainty and determine probabilities for the number of newspapers wanted on a particular day. If so, an example of a discrete descriptive problem is to compute the number of newspapers the news vendor must buy so there is at least a 95% probability that all will be sold that day. A continuous descriptive problem is to determine the probability the news vendor has unsold newspapers at the end of the day if she buys 25 newspapers. Another example is to find the average time it will take to sell 10 papers if the time between sales is random with a given distribution. If an airplane has 50 seats, an airline wants to know the probability that more than 50 people show up for the flight if 55 tickets are sold.

Compartment 7. SPPD (Static, Probabilistic, Prescriptive, Discrete).

Compartment 8. SPPC (Static, Probabilistic, Prescriptive, Continuous). Prescriptive problems seek the best answer among a number of alternatives. The vendor wants to know how many papers to purchase to maximize expected profit given a distribution for daily demand. Linear programs with random coefficients are an example of a continuous problem in this compartment. Decision analysis under risk or uncertainty [16] and stochastic programming [26, 27] are probabilistic prescriptive tools for Compartments 7 and 8. Monte Carlo techniques can also be used.

The remaining compartments are dynamic; the model parameters or outputs change, typically over time.

Compartment 9. DDDD (Dynamic, Deterministic, Descriptive, Discrete).

Compartment 10. DDDC (Dynamic, Deterministic, Descriptive, Continuous). Physics examples of continuous systems are falling bodies, spring systems, and voltage across capacitors or current through inductors in electrical circuits. Calculus and differential equations are often-used tools for these

systems. Fourier and Laplace transforms can also be helpful. These systems are deterministic. Engineers and scientists planning a space mission can predict where the planets will be years in advance; there is no uncertainty.

In discrete-time systems, time passes in discrete increments (e.g., whole hours or days). Discrete-time systems are common if digitized signals are involved. For example, a digital cell phone samples the speaker's voice 8000 times per second. Each sample results in an 8-bit codeword that is transmitted through the telephone network. The voice information is converted from analog (continuous information) to digital (discrete information). Since the information is periodic (one codeword every 1/8000 seconds), the network is synchronized with high-speed transmission channels interleaving codewords from hundreds or thousands of conversations. The listener's phone receives a codeword 8000 times per second. The phone converts the codeword (discrete information) into an analog (continuous information) signal that is (usually) heard by the listener. Compact disc and DVD players also receive periodic (i.e., one codeword every fixed time interval) digital data from the disc, which is converted into an analog signal for playback. Coding systems often attempt to predict the next codeword to reduce the number of bits needed to carry information. This prediction is an example of a dynamic, deterministic, descriptive discrete system.

Difference equations, discrete-time Fourier transforms, and z-transforms are primary tools for discrete-time systems [28, 29]. Electrical engineers will become intimately familiar with these in courses on digital signal processing, digital communications systems, and digital control systems.

Compartment 11. DDPD (Dynamic, Deterministic, Prescriptive, Discrete).

Compartment 12. DDPC (Dynamic, Deterministic, Prescriptive, Continuous). Continuous and discrete-time control systems are examples of these compartments [29, 30]. Consider a missile locked onto a target. The target may move and winds may deflect the missile. A prescriptive solution that changes over time is needed to guide the missile to its target. Another example is an automobile cruise control. A target speed is set and the throttle is adjusted to meet the target. Hills will affect the speed and the throttle must adjust to keep the target speed. These types of systems can be modeled in continuous time or, using sampling and digital electronics, in discrete time.

Compartment 13. DPDD (Dynamic, Probabilistic, Descriptive, Discrete).

Compartment 14. DPDC (Dynamic, Probabilistic, Descriptive, Continuous). Stochastic or random processes [23], discrete-event simulation [31] and time series analysis [32] are primary tools for systems with randomness that changes over time. These dynamic probabilistic systems are all around us: the number of people in line at the coffee shop throughout the day, the gas mileage over different segments of a trip home, whether or not the next person through the checkpoint is a smuggler. A company designing a new factory may wish to determine where the bottlenecks for material flow will be or predict the pick time for orders if the layout of a warehouse is

changed. A continuous-time model represents the system at all points in time. A discrete-time model may only represent the system at fixed points in time (e.g., quarterly).

Compartment 15. DPPD (Dynamic, Probabilistic, Prescriptive, Discrete).

Compartment 16. DPPC (Dynamic, Probabilistic, Prescriptive, Continuous). A call center wants to determine staffing levels so there is a 95% probability an incoming call is answered by a human operator. Call volume changes over time and the company wastes money if too many operators are on duty, so the optimal number of operators changes over time.

The price of an airline ticket changes over time. The airline wants to maximize revenue while selling a perishable commodity; after the flight leaves, an open seat has no value. Suppose an airline has 100 seats on a flight from New York to Phoenix departing at 9:20 a.m. next February 5th. Consumer demand for tickets that will occur 8 months prior to departure will be different than the demand that will occur 1 month prior. Demand fluctuates over time, but can be influenced by price. If the airline sets the price too low, all the seats are sold months in advance. If the price is too high, the flight leaves with many empty seats. Neither solution maximizes revenue. The airline needs a pricing strategy that sets the best ticket price to maximize revenue conditioned on the time remaining until the flight departs and the number of seats that have been sold at different pricing levels. Finding the optimal price for each ticket over time is an example of a dynamic, probabilistic, prescriptive problem. This is a yield management problem and is common in the travel industry for companies such as airlines, hotels, rental car operators, and cruise lines. A model that sets the ticket price every day would be a discrete-time model; one that sets the price at the time a quote is requested would be a continuous-time model.

Tools for these compartments include probabilistic dynamic programming [16] and Markov decision processes [23]. Metaheuristic optimization techniques can also be used with stochastic descriptive models [21].

This section presented the "big picture" of modeling. It began with a discussion of what models are and why they are used. Qualities were introduced as a way to evaluate or compare models. A sample modeling process was provided that included the different aspects of building a model. Finally, model characteristics were used to classify different types of models.

It is important to remember that, just as "a map is not the territory it represents," a model is not the system it represents—"all models are wrong." They have limitations, drawbacks, and faults, and unpleasant consequences can result if they are used haphazardly.

4.6 SIMULATION MODELING

The last section discussed how models can be relatively simple yet powerful tools often used by systems engineers. In the next section we will discuss another set of

tools used by systems engineers and which employ models: simulations. If models are analogous to claw hammers used to pound in nails or screw drivers to install screws, then simulations can be thought of as nail guns and drills which allow a carpenter to drive in numerous nails and screws much more rapidly. As we will see, each set of tools, models and simulations, has its appropriate place in the systems engineer's toolbox.

In its simplest form, a *simulation* is a model operated over time. In more mathematical terms, it is the numerical execution of a model to see the potential effect of the variables in question on the model's output [31]. In other words, a simulation is the execution of a model over time to produce varied outputs for use in analysis of the system.

A simulation is a model operated over time.

We use simulations to see the potential effects of time or some other variable(s) on the system a model is representing. Often, vast amounts of data representing the states of a system need to be generated in order to determine the statistical significance of change on the system. This data may be too difficult or costly to collect on the actual system in abundance to calculate the most probable effects of modifications to the system. A simulation of a viable model of a system can be used to generate this data.

4.6.1 Analytical Solutions versus Simulation; When It Is Appropriate to Use Simulation

It is appropriate to use simulation when a system is sufficiently complex that the possibility of a simple analytical solution is unlikely [31]. Many times we model systems which on the surface appear simple. However, once we begin to alter variable values, the mathematical representation or model becomes too complex to solve using a closed form analytical solution.

In general, choosing to use a simulation is based on six factors [33]:

- An operational decision based on a logical or quantitative model is needed.
- The system being analyzed is well-defined and repetitive.
- System/model activities, variables, and events are interdependent and change over time or condition.
- Cost/impact of the decision is greater than the cost of developing the model, executing the simulation, and assessing the data.
- Experimenting with the actual system costs more than developing the model, executing the simulation, and assessing the data.
- System events occur infrequently, not allowing for the system to achieve a steady state of performance where multiple adjustments to the system events can be made and assessed in an attempt to create a more efficient system.

The first three factors address system characteristics and structures. The next two address resource concerns. The last factor deals with the frequency of system usage. The first five factors can be used to help determine if a simulation is appropriate for assessing the alternatives generated in the solution design phase of the SDP. The last factor is a possible mitigating factor used to balance the risk of making or not making changes to a system which is used infrequently.

4.6.2 Simulation Tools

Analytical Solutions versus Simulation Simulation tools are categorized in different ways depending on their intended use. In this section, we will present both industry and military simulations to provide the broader scope of preexisting simulation tools, which currently exist and need not be developed. They do require the development of the scenarios but the basic simulation engine exists alleviating the need to program a theoretical model. Also, their user interfaces allow the rapid development of a simulation scenario. From industry, we will limit our discussion to ProModel®. With regard to military simulations, we will describe various military categories of simulations and provide a general overview of some of the existing simulations used in the analytical community.

Complex Queuing Simulations (ProModel®) ProModel® is a discrete-event simulation tool used by industry to model manufacturing, transportation, logistics, and service related systems. It allows systems to be represented as a series of queues and servers in which an entity, typically a customer product being produced/processed, is either being serviced or awaiting service. The simulation tool allows one to test various alternative layout designs and service processes prior to implementing the layout or process in the system. This tool has the fidelity to model "resource utilization, production capacity, productivity, inventory levels, bottlenecks, throughput times, and other performance measures" [33].

ProModel® is used by many industries to simulate their manufacturing and/or transportation system(s). ProModel® can also be used to simulate military systems. Figure 4.10 is an example of a student-lead systems engineering project to simulate alternative solutions for a proposed missile defense system. Similar simulation tools currently available on the market are Arena® and AutoMod.

Military Simulations *Military simulations* are normally used by the U.S. Department of Defense to simulate a myriad of aspects related to military operations and systems. These include, but are not limited to, issues related to: environmental factors, vehicle performance, communications capabilities, weapons effects, human behavior, and so on. There are fundamentally four categories of military simulation: simulation typologies, real-time versus non-real-time simulations, hierarchy of models and simulations, and military simulations versus games. These categories are illustrated in Figure 4.11.

Simulation Typologies. There are four *simulation typologies*: live, virtual, constructive, and augmented. The right side of Figure 4.11 is a visual depiction

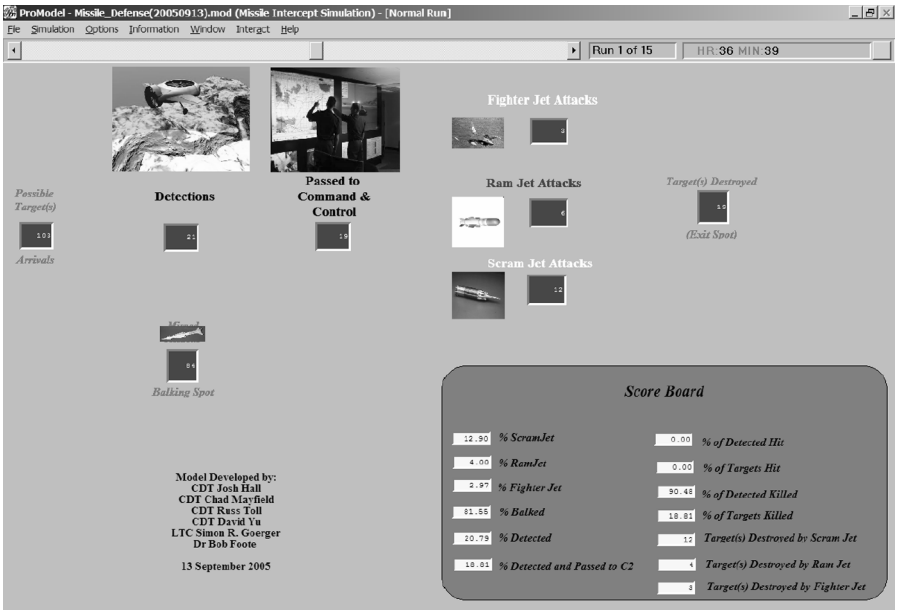

Figure 4.10 ProModel® anti-ballistic missile simulation example [34].

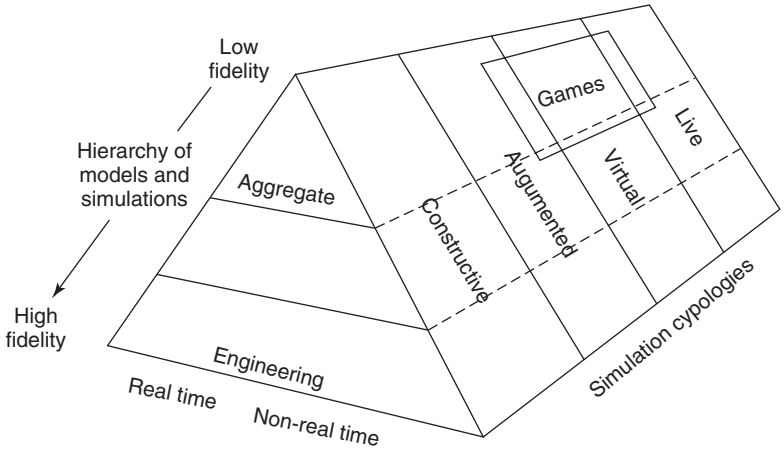

Figure 4.11 Simulation types [35].

of the relationship between the first three simulation typologies and games. *Live simulations* involve real people using real systems in a simulated environment. An example of a live simulation is an emergency response team conducting a full dress rehearsal on the launch pad prior to the launch of a manned rocket. *Virtual simulations* include the additional complexity of real users interacting with the simulated equipment. An astronaut conducting

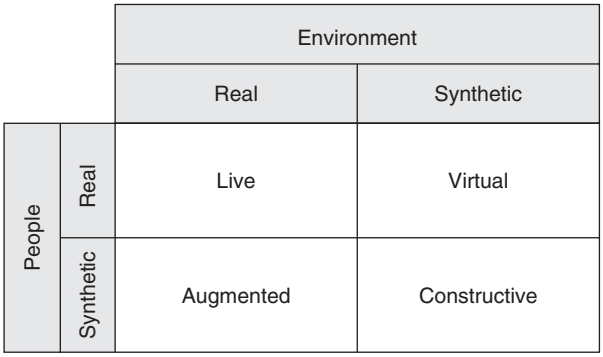

Figure 4.12 Simulation–reality relationships.

flight training in a simulator is an example of a virtual simulation. Each typology can be considered in terms of people and environment. Each may be either real or synthetic, as shown in Figure 4.12. A common augmented reality technique is to display computer-generated people in optical systems through which real people look. This allows the use of real weapons against synthetic criminals or combatants in a training environment without the risk of bodily harm or death.

The third typology is *constructive simulation*. In this typology, simulated entities operate in a simulated environment. Constructive simulations are normally used to simulate activities in rapid succession in order to provide insight into emerging system behavior or possible outcomes based on changes in equipment or processes. Agent-based simulations (see Section 4.6.2) of human behavior or multifaceted relationships between entities and which replicate hundreds, if not thousands, of entities interacting in a complex virtual environment are examples of constructive simulations. For example, NASA could use a constructive simulation to assess the effect(s) of asteroid activity on a space station. It could use other constructive simulations to predict the oxygen exchange rate in a space station based on the predicted growth of vegetation in the station, air filtering capacity of the ventilation system, and the number of personnel in the station.

Real-Time versus Non-Real-Time Simulations. Simulations can be categorized based on their ability to provide "real-time" performance or "non-real-time" performance. *Real-time* simulations are usually associated with virtual simulations that provide human participants with timely information, reactions, and effects. *Non-real-time* simulations are usually associated with constructive simulations that run uninterrupted for their duration. During solution design and decision-making phases of the SDP, systems engineers frequently use non-real-time simulations to generate data for use in analysis of alternatives. The use of non-real-time simulations allows users to conduct studies using the highest fidelity algorithms available with little concern for how long it takes to run the simulation because the system activities modeled in

this type of simulation are not constrained to actual clock time. Non-real-time simulations can work faster or slower than actual clock time. A real-time simulation could be used to conduct flight training or to test the reaction times of emergency response teams. A non-real-time simulation could be used to identify the disturbance of the air flow (lift and drag) as it streams over the fins of a new rocket design.

Hierarchy of Models and Simulations. Military simulations are also categorized according to the simulation's level of *fidelity* (see left side of Figure 4.11). The greater the fidelity, normally the greater the detail of the item(s) you are attempting to simulate. For example, a low-level fidelity model of a rocket system might represent the rocket as a solid object where a higher fidelity model might include the fins, fuel system, guidance system, and load capacity of the rocket. Fidelity can be seen as a continued scale but is routinely organized into smaller groupings to make it easier to group simulations into consistent hierarchy categories. Some organizations place simulations into a hierarchy consisting of five levels: campaign, theater, mission, engagement, and engineering [36]. The Army Modeling and Simulation Office's publication, "Planning Guidelines for Simulation and Modeling for Acquisition Requirements and Training," describes a four-level hierarchy where the hierarchy combines theater and campaign into one level [37]. Hughes refers to these four levels as campaign or theater, battle or multi-unit engagement, single engagement, and phenomenological [38].

The front triangle of Figure 4.11 is an illustration of a three-tier *hierarchy*: aggregate, entity, and engineering. The *aggregate* level consists of simulations or models that combine entities into groups based on functionality or association. Examples of these are boxes of gages or reels of wire in a rocket manufacturing plant warehouse or the payload capacities of the rockets, respectively. The aggregation of entities helps to reduce the computational requirements for modeling and simulating large-scale scenarios. Aggregate simulations routinely have less fidelity than is normally found at the entity and engineering level simulations. Aggregate simulations would represent a group of vehicles, such as a convoy, as it moves packaged goods from one side of a country to another.

Entity-level simulations reside between aggregate and engineering level simulations. Entity-level simulations represent individual platforms and the effects created by or acting on them. An entity is a platform, system, product, or individual that is creating effects or receives the effects of the system, such as the entity in ProModel®. As such, a "distinguishable person, place, unit, thing, event, or concept" simulation must maintain or track information about each entity in the model [39]. An entity-level simulation of an asteroid field would replicate each individual asteroid, its relevant behaviors, and positional data as a rocket passes through the asteroid field.

Engineering-level simulations normally deal with the components of one or two individual systems or subsystems. These simulations may go into more

detail (higher fidelity) to model the physical aspects of a system. Modeling the subcomponents, components, and behaviors of a rocket's guidance system is an example of an engineering-level simulation. Organizations use this form of modeling and simulation prior to building a new piece of equipment or integrating new technologies into existing facilities or systems before testing the changes using live simulations.

Systems engineers choose the appropriate level of simulation fidelity based on what aspect(s) of a system they are assessing.

Simulations versus Games. The military and industry produce numerous simulations to provide training for their people or to conduct analysis of operations/systems. The needs of the research and training communities place different constraints on the validity of the model or simulation. However, in the gaming industry the overriding concern is to manufacture a product people are interested in purchasing and playing. A notable exception is the game America's Army™, which was originally built as a recruiting tool for the U.S. Army.

Games are "activit(ies) engaged in for diversion or amusement" [40]. As such, the focus of the gaming industry has been to produce simulations which deliver entertainment value to its customers; however, developing or adapting games for use in education or training is becoming more popular. The belief that individuals will learn more if they are actively engaged in the learning activity is one of the fundamental reasons for using games. Most computerized games are simulations primarily designed for entertainment but may be used for learning, individual or team training, or analytical studies seeking to gain insight into human behavior.

One can categorize games as aggregate or entity-level simulations. Aggregate games are strategic in nature (e.g., Risk®, Axis & Allies, Railroad Tycoon™, Kohan II: Kings of War, etc.). In contrast, first shooter and role-playing games can be considered entity-level games (e.g., America's Army®, Air-Sea Battle, Alex Kidd in the Enchanted Castle, etc.). No matter which category of simulation a game resides, game developers' primary concern is entertainment not realism or the accurate portrayal of human performance during the design and coding of a game. This lack of realism can place games at the lower end of the fidelity spectrum (see Figure 4.11).

Simulation Behaviors A simulation executes based on the prescribed set of *behaviors* outlined for each entity. These behaviors can be complex or very simple. They can be reactive or proactive in nature. Simulation behaviors are most often limited based on the model architecture used to represent these behaviors. Generally speaking, the five most popular cognitive model representations in use in the late twentieth and early twenty-first centuries are agent-based, Bayesian-network, multiagent system, neural-networks, and rule-based. We will only address rule-based, agent-based, and multiagent representations in this section.

Rule-Based Simulations. A *rule-based* (knowledge-based) simulation replicates human behavior using a catalog of actions with causal if/then association to select and execute an appropriate action [41, 42]. This causal representation often requires an extensive effort to identify and code all relative possible conditions an entity may encounter along with viable entity actions for those conditions. Subject matter experts are routinely used to establish and validate these data prior to its use. Rule-based simulations are best used to model systems, which are physics-based, or for replicating systems with a relatively limited (countable) number of states and actions.

Agent-Based Simulations. Agent-based representations model intelligence through codified objects that perceive characteristics of the environment and act on those perceptions [41]. There are several types of agent-based cognitive architectures. Two of these are *reactive* and *rational* agents. A reactive agent bases its actions solely on the last set of sensory inputs. Often the approach uses a simple condition action rule (e.g., if this is my perceived state of world, then I choose this action). A rational agent uses sensors to perceive its environment and performs actions on the environment using effectors. Rational agents maintain a state of situational awareness based on their past knowledge of the world and current sensory inputs [41].

Multiagent System Simulations. The *multiagent system* (MAS) is a relatively new representation for modeling and simulating behaviors based on the complex adaptive system (CAS) theory. Developed in the late 1970s, MAS is a system with autonomous or semiautonomous software agents that produce adaptive and emergent behaviors. The model uses a bottom-up approach where software agents have independent micro decisions that generate group and system-level macro behaviors. A MAS can use any form of agent-based software technology (reactive, rational, goal-based, utility-based, etc.) that has agents characterized as possessing intentions that influence their actions. Multiagent systems are used in large domains where nonlinearity is present [43]. The MAS, limited only by the physics constraints of the simulation boundaries, uses an indirect approach to search the large domain for viable results. Another feature of MAS is its ability to allow agents to evolve to create new agents which, in general, are better suited to survive or prosper in the simulated environment [44].

Agent-based and MAS simulations are often used to explore a wide spectrum of possible system effects based on an extensive range of variables inputs. This allows systems engineers to assess the likelihood of possible system-level behavior in the context of various system constraints. Agent-based tools often lack the higher fidelity levels found in most physics-based simulations. However, these lower fidelity agent-based models allow for more varied behavior interactions. This makes them highly useful for simulating system alternatives in which complex, nonlinear interactions can occur between entities that are difficult, if not impossible, to specify in the closed-form expressions required of physics-based models and simulations.

4.7 DETERMINING REQUIRED SAMPLE SIZE

A convenience of deterministic models is that given a known set of inputs, the same output is ensured. For example, using $E = mc^2$ with a given mass, if the speed of light was unknown or uncertain and consequently represented by a random variable between 299 and 300 million meters per second(mmps), instead of a constant 299,792,458 mmps, then a single calculation would not necessarily predict the true amount of available energy. Instead, some number of calculations, each with a different value for the random variable, would be required to gain a certain level of confidence that the true value had been captured. These calculations, interchangeably called *replications*, *trials*, or *runs*, may significantly add to the cost of an experiment, so the modeler needs a mechanism to understand the trade-offs between replication cost and certainty. Deterministic models of complex systems may also benefit from testing only a sample of the possible combinations instead of the entire population.

Two methods are usually used to determine a reasonable number of replications, the *required sample size*, on which to base an estimate. The first, *power analysis*, is an approach that determines how much "power" a test has to detect a particular effect. It is often referred to as the *power of the test*. This approach is not explored in this chapter.

The second method is based on the logic of a statistical confidence interval (CI). In this case, the goal is to have a certain degree of confidence that the true population mean has been captured by determining the sample mean, given a certain number of observations and an acceptable probability of error. The formula for determining a confidence interval is shown in Equation (4.3) as

$$CI_n = \bar{x}_n \pm t_{n-1,1-\alpha/2}\sqrt{\frac{s_x^2}{n}}. \tag{4.3}$$

This says that we can believe, with only an α probability of error, that the true mean is the same as the sample mean, $\bar{x}_n$, give or take some margin of error. The margin of error—everything to the right of the $\pm$ symbol—is made up of the variance of the sample, s_x^2, a desired level of confidence, $1 - \alpha$, and the t statistic for $n - 1$ degrees of freedom where n is the sample size.

For example, a health inspector may take 10 samples of drinking water to measure the amount of a contaminant and records the following levels, in parts per million (ppm):

$$29, 34, 20, 26, 20, 35, 23, 30, 27, 34.$$

The mean, $\bar{x}_n$, is found to be 27.8; the variance, s_x^2, is 31.51; n is 10; and α is 0.10 for a 90% confidence interval. Inserting these numbers into Equation (4.3), we can expect the true mean value of contaminant in the drinking water to be 27.8 ± 3.254, or an interval of between 24.546 and 31.054 ppm.

Since everything to the right of the $\pm$ symbol represents margin of error, the number of samples required, n, can be calculated if the modeler is willing to specify

that error up-front. This is done routinely, for example, when someone declares some value, "give or take a couple." This is the same approach taken in power analysis when the modeler selects a level of effect to be detected. The modeler is saying, "I want to know how many samples/trials/replications/runs are required to detect a certain margin or error." These calculations can be done easily in a spreadsheet, although the modeler must rely on experience and judgment to avoid certain pitfalls. Specifically:

- Select a meaningful measure to use for sample size estimation.
- Generate a reasonable number of pilot runs to establish a solid mean and variance.
- Beware of the assumption that the pilot variance represents the true variance.
- Always round up the estimated n to the next higher number. Err on the side of caution.
- Generate an estimated n from the pilot runs, then incrementally increase the actual runs, rechecking the estimate for n, until the estimated n equals the actual n and your professional judgment is comfortable with the results and they are defensible to a decision maker.
- The larger the n, the greater the likelihood that the sample mean represents the true mean. Find the balance between replication cost and result accuracy.

Consider a golf club manufacturer interested in the performance of a new club. A swing machine hits 10 balls that fly the following distances, in yards:

$$220, 210, 189, 201, 197, 200, 205, 198, 196, 200$$

Engineers want to use this series of pilot replications to determine how many balls need to be hit to find the mean distance, plus or minus two yards. By saying that, they acknowledge that they are willing to have a margin or error of two yards. Since it has been shown that the margin of error is represented in Equation (4.4), the engineers can set the margin of error equal to two and solve for n, as in Equations (4.5) through (4.9):

$$\text{Margin of error} = t_{n-1,1-\alpha/2}\sqrt{\frac{s_x^2}{n}} \tag{4.4}$$

$$2 = 1.833\sqrt{\frac{72.27}{n}} \tag{4.5}$$

$$\Rightarrow 2 = \frac{(1.833)\,(8.5)}{\sqrt{n}} \tag{4.6}$$

$$\Rightarrow \sqrt{n} = \frac{(1.833)\,(8.5)}{2} \tag{4.7}$$

$$\Rightarrow n = \left(\frac{15.58}{2}\right)^2 \tag{4.8}$$

$$\Rightarrow n = 60.68 \tag{4.9}$$

Hitting 61 more golf balls is not likely to be a problem, unless they replace the swing machine with a professional golfer like Tiger Woods, whose per-swing fee may be very costly to the company. In that case, they may have Tiger hit another 10 balls and recalculate the required n. They can continue to incrementally move toward the 61-ball goal until the revised n equals the actual n. With each increment, the recorded variance approaches the true variance for the club and is therefore a better representation of reality.

Although this approach provides only an estimate, it also provides the modeler with an easy and defensible tool for determining how many replications are required to achieve a desired result.

4.8 SUMMARY

Modeling a system's essential components, attributes, and relationships, and understanding how they change over time, provides systems engineers with critical insights into all stages of the system's life cycle. While Chapter 2 appropriately frames the system thinking perspective on how systems can be thought of and represented, this chapter has introduced tools that reveal the inner workings of a system. Chapter 4 highlights the fundamentals of system measures, models, simulations, and required sample size to an understanding of the life cycle. It starts with three key reasons for understanding the system life cycle, all of which are supported directly with the tools and techniques from this chapter. The three motivating reasons are as follows:

1. We can organize system development activities in a logical fashion that recognizes some activities must be accomplished prior to others.
2. We can identify the specific activities needed to be accomplished in each stage to successfully move to the next stage.
3. We can effectively consider the impact that early decisions have on later stages of the systems life cycle, especially with regard to cost and risk.

Knowing which tools to use at given stages of the system life cycle is part of the art of systems engineering illustrated throughout this book. There is rarely a single right answer, though there are always many wrong ones. The system modeler must always keep in mind that as abstract representations of objects or processes, models at best provide only insights into understanding a system and at worst lead decision makers astray. The opening quote of this chapter, "All models are wrong, some are useful," must never be far from the systems engineer's mind.

4.9 EXERCISES

4.1. Use model qualities to compare and contrast the following models used at a fast food restaurant to order burger supplies for the next week:

(a) Fit a distribution to weekly data from the past year, then order an amount that would satisfy demand for 95% of the weeks.

(b) Order supplies based on how many burgers were sold last week.

4.2. Use model qualities to compare and contrast the following models used to predict gas mileage for an automobile:

(a) Randomly choose a number of different destinations within 100 miles of your present location. Gather experimental data by driving from your location to each destination and back. Record the average speed and average mileage. Fit a line to the data.

(b) Develop a formula to predict gas mileage by using formulas for kinetic energy and chemical–mechanical conversion with assumptions about wheel bearing friction and engine torque.

4.3. Describe an example of a robust and a nonrobust system or model. Justify your labels.

4.4. Describe an example of a high-fidelity and a low-fidelity system or model. Justify your labels.

4.5. Describe an example of a system or model that is dynamic and deterministic.

4.6. Describe an example of a system or model that is deterministic and probabilistic.

4.7. Describe an example of a system or model that is static and predictive.

4.8. Describe an example of a system or model that is descriptive and dynamic.

4.9. A security checkpoint screens passengers entering the passenger terminal of an airport. List the probabilistic elements in the system. Describe how you could make these elements deterministic.

4.10. For each of the following problems, categorize the following system or model using each of the four characteristics (i.e., the modeler's toolbox).

(a) A rubber duck floating in a bathtub.

(b) A NASCAR pit stop.

(c) The latest version of the computer game Halo.

(d) $E = mc^2$.

(e) The Illinois Lottery Mega Millions game.

(f) The Federal Emergency Management Agency (FEMA) wishes to develop a model to predict the resiliency of a metropolitan area; how quickly an area can return to normal following a natural or man-made disaster.

(g) A state department of transportation wants to determine how long the left-turn lane should be at a busy intersection.

(h) An artillery officer needs to determine the angle of launch for a given type of artillery round.

(i) The U.S. Air Force must determine which planes to fly which routes to which destinations. Each type of plane has its own range, speed, load size, and weight capabilities. The location of supply bases and the locations of receiving bases and their materiel requirements are given.

(j) The FEMA is pre-positioning disaster relief supplies in regional areas. They need to determine where these supplies should be located so they can be quickly deployed in an emergency. The likelihood of different types of disasters (e.g., floods, hurricanes, earthquakes, tornadoes) and their severity must be considered. The budget is limited and FEMA wants to maximize the impact they have following a disaster.

(k) An investment company has $1 million to invest in five different asset classes (e.g., short-term treasury notes, stocks, bonds, etc.). The company must consider the return and risk of each asset class. Return is usually measured as the expected value of the return over some time horizon (e.g., a year). Risk is usually measured as the standard deviation of return over the same time horizon. The company wants to maximize its return while limiting risk to a set level.

4.11. What is the difference between a model and a simulation?

4.12. Why would a systems engineer use a model?

4.13. List and describe the four qualities you feel are the most important for a model to be useful. Why did you pick these four?

4.14. List and describe the three types of models. Give examples of each (different than those found in this chapter).

4.15. What are the four major steps in the modeling process?

4.16. List the three means of exercising a model and explain how they differ.

4.17. What is the difference between a deterministic and a probabilistic model? Give an example of each using a fast-food restaurant as your system.

4.18. When and why is it appropriate for a systems engineer to use a simulation?

4.19. List and describe the three simulation typologies. Give an example of each using a fast-food restaurant as your system.

4.20. Which of the three simulation hierarchies would be used to describe a model of traffic flow of individual vehicles down an interstate highway? Why?

4.21. What is the difference between a game and a simulation? Can a game be a simulation? Why?

4.22. Construct a seven-paragraph measure of effectiveness for an unmanned aerial vehicle (UAV) designed for U.S. border surveillance.

4.23. Pilot runs of a simulated emergency response system results in response times of 25, 42, 18, 36, 28, 40, 20, and 34 min. Calculate the required sample size necessary to be 90% confident that the sample mean represents the true mean, given or take 3 min.

REFERENCES

1. Sage, AP, Armstrong, JE, Jr. *Introduction to Systems Engineering*. Wiley Series in Systems Engineering. New York: Wiley-Interscience, 2000.
2. *The Measures of Effectiveness*, USACDC Pamphlet No. 71–1. U.S. Army Combat Developments Command, Fort Belvoir, VA, 1973.
3. http://www.wrightflyer.org/WindTunnel/testing1.html. Accessed April 4, 2010.
4. Porter, N. 1996. The Wright Stuff. WGBH Boston. Available at http://www.lilienthal-museum.de/olma/ewright.htm. Accessed April 4, 2010.
5. http://www.importanceofphilosophy.com/Esthetics_Art.html. Accessed April 4, 2010.
6. http://pespmc1.vub.ac.be/OCCAMRAZ.html. Accessed April 4, 2010.
7. Simulation Interoperability Standards Organization (SISO), Fidelity Implementation Study Group (ISG). Fidelity ISG Glossary. V 3.0, 16 December 1998.
8. "A Glossary of Modeling and Simulation Terms for Distributed Interactive Simulation (DIS)," Office of the Under Secretary of Defense (Acquisition and Technology), Washington, DC. August, 1995.
9. Department of Defense Instruction. 2003. DoD Instruction 5000.61: DoD Modeling and Simulation (M&S) Verification, Validation and Accreditation (VV&A). Available at https://www.dmso.mil/public/library/policy/policy/i500061p.pdf. Accessed June 7, 2006.
10. Aaronson, S. Guest column: NP-complete problems and physical reality. *ACM SIGACT News Archive*, 2005;36(1):30–52.
11. Ragsdale, C. *Spreadsheet Modeling and Decision Analysis*, 4th ed. Mason, OH: South-Western College Publishing, 2003.
12. Foulds, LR. *Combinatorial Optimization for Undergraduates*. Undergraduate Texts in Mathematics. New York: Springer, 1984.
13. Wolsey, L. *Integer Programming*. New York: Wiley-Interscience, 1998.
14. Institute for Operations Research and the Management Sciences. Available at http://www2.informs.org/Resources/. Accessed April 4, 2010.
15. Powell, SG, Baker, KR. *The Art of Modeling with Spreadsheets*: *Management Science, Spreadsheet Engineering, and Modeling Craft*. New York: John Wiley & Sons, 2003.
16. Hillier, FS, Lieberman, GJ. *Introduction to Operations Research*, 8th ed. New York: McGraw-Hill, 2005.
17. Taha, HA. *Operations Research*: *An Introduction*. 6th Ed. Upper Saddle River, NJ: Prentice-Hall, 1996.
18. Winston, WL, *Operations Research*: *Applications and Algorithms*, 4th ed. Belmont, CA: Duxbury Press, 2003.
19. Gill, P, Murray, W, Wright, MH. *Practical Optimization*. New York: Academic Press, 1981.
20. Nocedal, J, Wright, S. *Numerical Optimization*. New York: Springer-Verlag, 1999.
21. Glover, FW, Kochenberger, GA. *Handbook of Metaheuristics*. International Series in Operations Research & Management Science. New York: Springer, 2003.
22. Ross, S. *A First Course in Probability*, 6th ed. Upper Saddle River, NJ; Prentice-Hall, 2001.
23. Ross, S. *Introduction to Probability Models*, 8th ed. New York: Academic Press, 2003.

24. Devore, J. *Probability and Statistics for Engineering and the Sciences*, 6th ed. Belmont, CA: Duxbury Press, 2004.
25. Hayter, A. *Probability and Statistics for Engineers and Scientists*, 2nd ed. Belmont, CA: Duxbury Press, 2002.
26. Birge, JR, Louveaux, F. *Introduction to Stochastic Programming*. New York: Springer, 2000.
27. Spall, JC. *Introduction to Stochastic Search and Optimization*. New York: John Wiley & Sons, 2003.
28. Proakis, JG, Manolakis DG. *Discrete Signal Processing, Principles, Algorithms, and Applications*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.
29. Franklin, GF, Powell, JD, Workman, ML. *Digital Control of Dynamic Systems*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1997.
30. Norman, S, Nise, NS. *Control Systems Engineering*, 4th ed. New York: John Wiley & Sons, 2003.
31. Law, AW, Kelton, WD. *Simulation Modeling and Analysis*, 3rd ed. New York: McGraw-Hill, 2000.
32. Chatfield, C. *The Analysis of Time Series*: *An Introduction*, 6th ed. Boca Raton, FL: Chapman & Hall/CRC, 2003.
33. Harrell, C, Ghosh, BK, Bowden, RO. *Simulation Using ProModel®*, 2nd ed. New York: McGraw-Hill, 2000.
34. Foote, BL, Goerger, SR. Design considerations for simulating ABM systems. Presentation to Huntsville Simulation Conference, Huntsville, Alabama, 2005.
35. Goerger S. R. Validating Computational Human Behavior Models: Consistency and Accuracy Issues. Doctoral dissertation, Department of Operations Research. Monterey, CA: Naval Postgraduate School, 2004.
36. Combat Modeling Overview. Class notes, Introduction to Joint Combat Modeling (OA/MV4655), Operations Research Department, Naval Postgraduate School, Monterey, California, July 2002.
37. Army Modeling and Simulation Office (AMSO) Planning Guidelines for Simulation and Modeling for Acquisition Requirements and Training (Change 1). September 15, 2002. Available at http://www.amso.army.mil/smart/documents/guidelines/guidelines-revisedsep02.doc. Accessed June 7, 2006.
38. Hughes, WP. *Military Modeling for Decision-Making*, 3rd ed. Alexandria, VA: Military Operations Research Society, 1997.
39. *Military Handbook for Joint Data Base Elements for Modeling and Simulation (M&S)*. August 5, 1993.
40. Merriam-Webster Online [WWW Document]. Available at http://www.m-w.com/cgi-bin/dictionary]. Accessed June 28, 2006.
41. Russell, S, Norvig, P. *Artificial Intelligence*: *A Modern Approach*. Upper Saddle River, NJ: Prentice-Hall, 1995.
42. Dean, T, Allen, J, Aloimonos, J. *Artificial Intelligence: Theory and Practice*. Redwood City, CA: Benjamin/Cummings Publishing Company, 1995.
43. Holland, JH. *Hidden Order*: *How Adaptation Builds Complexity*. Cambridge, MA: Perseus Books, 1995.
44. Freedman, A. *The Computer Desktop Encyclopedia (CD Rom—Ver. 12.1)*. Point Pleasant, PA: Computer Language Company, 1999.