

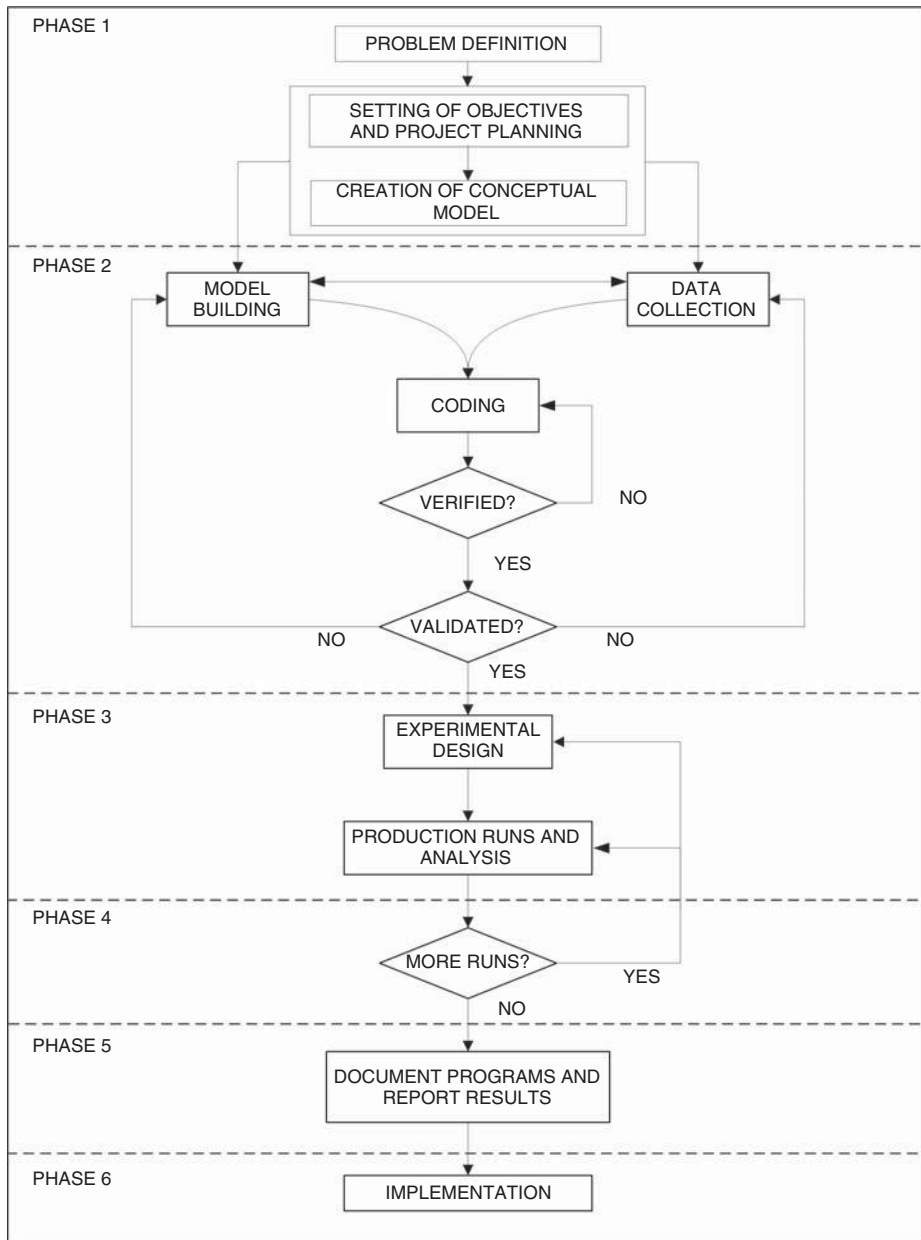
1. Define the problem.
2. Establish measures of performance for evaluation.
3. Generate alternative solutions.
4. Rank alternative solutions.
5. Evaluate and iterate during process.
6. Execute and evaluate the solution.

This methodology can be referred to by using the first letter of each step. The DEGREE methodology for problem solving represents a series of steps that can be used during the problem-solving process. The first step helps to ensure that you are solving the right problem. The second step helps to ensure that you are solving the problem for the right reason, that is, your metrics must be coherent with your problem. Steps 3 and 4 ensure that the analyst looks at and evaluates multiple solutions to the problem. In other words, these steps help to ensure that you develop the right solution to the problem. A good methodology recognizes that the analyst needs to evaluate how well the methodology is doing. In step 5, the analyst evaluates how the process is proceeding and allows for iteration. Iteration is an important concept that is foreign to many modelers. The concept of iteration recognizes that the problem-solving process can be repeated until the desired degree of modeling fidelity has been achieved. Start the modeling at a level that allows it to be initiated and do not try to address the entire situation in each of the steps. Start with small models that work and build them up until you have reached your desired goals. It is important to get started and get something established on each step and continually go back in order to ensure that the model is representing reality in the way that you intended. The final step is often overlooked. Simulation is often used to recommend a solution to a problem. Step 6 indicates that if you have the opportunity, you should execute the solution by implementing the decisions. Finally, you should always follow up to ensure that the projected benefits of the solution were obtained.

The DEGREE problem-solving methodology should serve you well; however, simulation involves certain unique actions that must be performed during the general overall problem-solving process. When applying DEGREE to a problem that may require simulation, the general DEGREE approach needs to be modified to explicitly consider how simulation will interact with the overall problem-solving process.

Figure 1.4 represents a refined general methodology for applying simulation to problem solving.

1. Problem formulation
  - (a) Define the problem
  - (b) Define the system
  - (c) Establish performance metrics
  - (d) Build conceptual model
  - (e) Document model assumptions
2. Simulation model building
  - (a) Model translation
  - (b) Input data modeling
  - (c) Verification
  - (d) Validation



**Figure 1.4** General simulation methodology

### 3. Experimental design and analysis

- (a) Preliminary runs
- (b) Final experiments
- (c) Analysis of results

4. Evaluate and iterate
  - (a) Documentation
  - (b) Model manual
  - (c) User manual
5. Implementation

The first phase, problem formulation, captures the essence of the first two steps in the DEGREE process. The second phase, model building, captures the essence of step 3 of the DEGREE process. When building models, you are either explicitly or implicitly developing certain design alternatives. The third phase, experimental design and analysis, encapsulates some of steps 3 and 4 of the DEGREE process. In designing experiments, design alternatives are specified, and when analyzing experiments, their worth is being evaluated with respect to problem objectives. The fourth phase, evaluate and iterate, captures the notion of iteration. Finally, the fifth and sixth phases, documentation and implementation complete the simulation process. Documentation is essential when trying to ensure the ongoing and future use of the simulation model, and implementation recognizes that simulation projects often fail if there is no follow through on the recommended solutions.

The problem formulation phase of the study consists of five primary activities:

1. Defining the problem.
2. Defining the system.
3. Establishing performance metrics.
4. Building conceptual models.
5. Documenting modeling assumptions.

A problem starts with a perceived need. These activities are useful in developing an appreciation for and an understanding of what needs to be solved. The basic output of the problem definition activity is a problem definition statement. A problem definition statement is a narrative discussion of the problem. A problem definition statement is necessary to accurately and concisely represent the problem for the analyst and the problem stakeholders. This should include all the required assumptions made during the modeling process. It is important to document your assumptions so that you can examine their effect on the model during the verification, validation, and experimental analysis steps of the methodology. This ensures that the problem is well understood and that all parties agree on the nature of the problem and the goals of the study. The general goals of a simulation study often include the following.

- *Comparison.* To compare system alternatives and their performance measures across various factors (decision variables) with respect to some objectives.
- *Optimization.* This is a special case of comparison in which you try to find the system configuration that optimizes performance subject to constraints.
- *Prediction.* To predict the behavior of the system at some future point in time.
- *Investigation.* To learn about and gain insight into the behavior of the system, given various inputs.

These general goals will need to be specialized to the problem under study. The problem definition should include a detailed description of the objectives of the study, the desired outputs from the model, and the types of scenarios to be examined or decisions to be made.

The second activity of this phase produces a definition of the system. A system definition statement is necessary to accurately and concisely define the system, particularly its boundaries. The system definition statement is a narrative, which often contains a pictorial representation of the major elements of the system. This ensures that the simulation study is focused on the appropriate areas of interest to the stakeholders and that the scope of the project is well understood.

When defining the problem and the system, one should naturally begin to develop an understanding of how to measure system performance. The third activity of problem formulation makes this explicit by encouraging the analyst to define the required performance measures for the model. To meaningfully compare alternative scenarios, objective and measurable metrics describing the performance of the system are necessary. The performance metrics should include quantitative statistical measures from any models used in the analysis (e.g., simulation models), quantitative measures from the systems analysis (e.g., cost/benefits), and qualitative assessments (e.g., technical feasibility, human, operational feasibility). The focus should be placed on the performance measures that are considered to be the most important to system decision makers and tied directly to the objectives of the simulation study. Evaluation of alternatives can then proceed in an objective and unbiased manner to determine which system scenario performs the best according to the decision maker's preferences.

The problem definition statement, the system definition statement, and explicit performance metrics set the stage for more detailed modeling. These activities should be captured in a written form. Within this text, you will develop models of certain "ready-made" book problems. One way to accomplish the problem formulation phase of a simulation study is to consider writing yourself a "book problem." You will need enough detail in these documents that a simulation analyst (you) can develop a model in any simulation language for the given situation. The example problem in Chapter 11 represents an excellent sample of problem and system definition statements. If you have the opportunity to do a "real-life" project as a part of your study of simulation, you might want to utilize the book problems in this text and the example in Chapter 11 for how to write reasonable problem/system definition statements.

With a good understanding of the problem and the system under study, you should be ready to begin your detailed model formulations. Model formulation does not mean a computer program. You should instead use conceptual modeling tools: conceptual diagrams, flow charts, etc. prior to any use of software to implement a model. The purpose of conceptual modeling tools is to convey a more detailed system description so that the model may be translated into a computer representation. General descriptions help to highlight the areas and processes of the system that the model will simulate. Detailed descriptions assist in simulation model development and coding efforts. Some relevant diagramming constructs include the following:

1. *Context Diagrams.* A context diagram assists in conveying the general system description. The diagram is a pictorial representation of the system that often includes flow patterns typically encountered. Context diagrams are often part of the system description document. There are no rules for developing context diagrams. If you have an artistic side, here is your opportunity to shine!

2. *Activity Diagrams*. An activity diagram is a pictorial representation of the process for an entity and its interaction with resources while within the system. If the entity is a temporary entity (i.e., it flows through the system), the activity diagram is called an activity flow diagram. If the entity is permanent (i.e., it remains in the system throughout its life), the activity diagram is called an activity cycle diagram. Activity diagrams will be used extensively within this text.
3. *Software Engineering Diagrams*. Because simulation entails software development, the wide variety of software engineering diagramming techniques can be utilized to provide information for the model builder. Diagrams such as flow charts, database diagrams, IDEF (ICAM Definition language) diagrams, UML (unified modeling language) diagrams, and state charts are all useful in documenting complex modeling situations. These techniques assist development and coding efforts by focusing attention on describing, and thus understanding, the elements within the system. Within this text, activity diagrams will be augmented with some simple flow chart symbols and some simple state diagrams will be used to illustrate a variety of concepts.

In your modeling, you should start with an easy conceptual model that captures the basic aspects and behaviors of the system. Then, you should begin to add details, considering additional functionality. Finally, you should always remember that the complexity of the model has to remain proportional to the quality of the available data and the degree of validity necessary to meet the objectives of the study. In other words, do not try to model the world!

After developing a solid conceptual model of the situation, simulation model building can begin. During the simulation model building phase, alternative system design configurations are developed based on the previously developed conceptual models. Additional project planning is also performed to yield specifications for the equipment, resources, and timing required for the development of the simulation models. The simulation models used to evaluate the alternative solutions are then developed, verified, validated, and prepared for analysis. Within the context of a simulation project, this process includes the following.

- *Input Data Preparation*. Input data is analyzed to determine the nature of the data and further data collection needs. Necessary data is also classified into several areas. This classification establishes different aspects of the model that are used in model development.
- *Model Translation*. Description of the procedure for coding the model, including timing and general procedures and the translation of the conceptual models into computer simulation program representations.
- *Verification*. Verification of the computer simulation model is performed to determine whether or not the program performs as intended. To perform model verification, model debugging is performed to locate any errors in the simulation code. Errors of particular importance include improper flow control or entity creation, failure to release resources, and logical/arithmetic errors or incorrectly observed statistics. Model debugging also includes scenario repetition utilizing identical random number seeds, “stressing” the model through a sensitivity analysis (varying factors and their levels) to ensure compliance with anticipated behavior, and testing of individual modules within the simulation code.
- *Validation*. Validation of the simulation model is performed to determine whether or not the simulation model adequately represents the real system. The simulation model

is shown to personnel (of various levels) associated with the system in question. Their input concerning the realism of the model is critical in establishing the validity of the simulation. In addition, further observations of the system are performed to ensure model validity with respect to actual system performance. A simple technique is to statistically compare the output of the simulation model to the output from the real system and to analyze whether there is a significant (and practical) difference between the two.

Model translation will be a large component of each chapter as you learn how to develop simulation models. Verification and validation techniques will not be a major component of this text, primarily because the models will be examples made for educational purposes. This does not mean that you should ignore this important topic. You are encouraged to examine many of the useful references on validation, see, for example, Balci [1997] and Balci [1998a].

After you are confident that your model has been verified and validated to suit your purposes, you can begin to use the model to perform experiments that investigate the goals and objectives of the project. Preliminary simulation experiments should be performed to set the statistical parameters associated with the main experimental study. The experimental method should use the simulation model to generate benchmark statistics of current system operations. The simulation model is then altered to conform to a potential scenario and is rerun to generate comparative statistics. This process is continued, cycling through suggested scenarios and generating comparative statistics to allow evaluation of alternative solutions. In this manner, objective assessments of alternative scenarios can be made.

For a small set of alternatives, this “one at a time” approach is reasonable; however, often there are a significant number of design factors that can affect the performance of the model. In this situation, the analyst should consider utilizing formal experimental design techniques. This step should include a detailed specification of the experimental design (e.g., factorial) and any advanced output analysis techniques (e.g., batching, initialization bias prevention, variance reduction techniques, and multiple comparison procedures) that may be required during the execution of the experiments. During this step of the process, any quantitative models developed during the previous steps are exercised. Within the context of a simulation project, the computer simulation model is exercised at each of the design points within the stipulated experimental design.

Utilizing the criteria specified by system decision makers, and utilizing the simulation model’s statistical results, alternative scenarios should then be analyzed and ranked. A methodology should be used to allow the comparison of the scenarios that have multiple performance measures that trade-off against each other.

If you are satisfied that the simulation has achieved your objectives, then you should document and implement the recommended solutions. If not, you can iterate as necessary and determine if any additional data, models, experimentation, or analysis is needed to achieve your modeling objectives. Good documentation should consist of at least two parts: a technical manual, which can be used by the same analyst or by other analysts, and a user manual. A good technical manual is very useful when the project has to be modified, and it can be a very important contribution to software reusability and portability. The approach to document the example models in this text can be used as an example for how to document your models. In addition to good model development documentation, often the simulation model will be used by non-analysts. In this situation, a good user manual for how to use and exercise the model is imperative. The user manual is a product for the user who may