

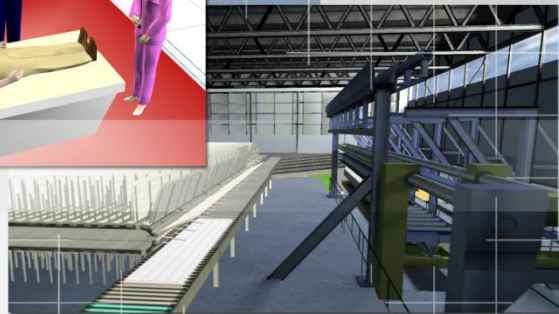
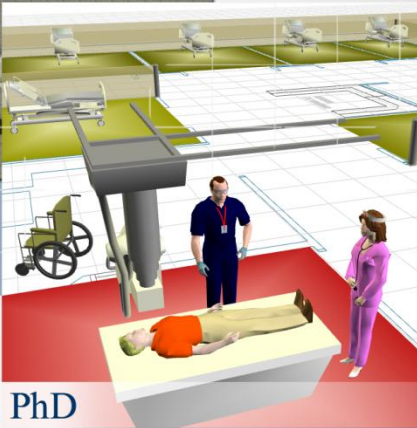
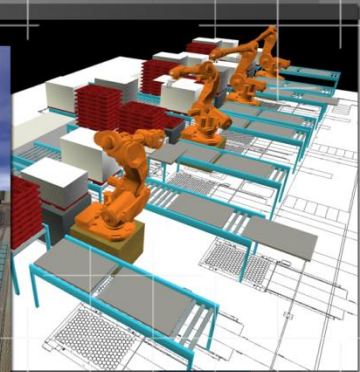
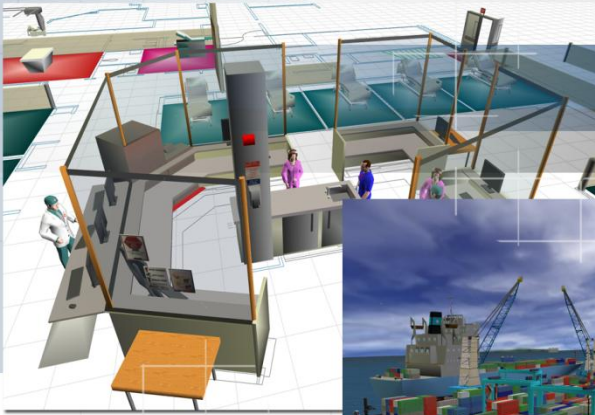
SELECTED EXCERPTS FROM

Applied Simulation

Modeling and Analysis

using **Flexsim**

3D Simulation Software



Malcolm Beaverstock, PhD
Allen Greenwood, PhD, PE
Eamonn Lavery, PhD
William Nordgren, MS CIM

visualizing and optimizing dynamic systems

Preface

Simulation is an applied technology that is especially useful for analyzing and solving problems. Applying simulation begins by being clear on the problem definition, the reasons for simulating, and the expected outcomes. Simulation with no objective is counterproductive.

A person using simulation then must balance their understanding of the problem with their knowledge of the details of simulation: the underlying simulation concepts, application software, and the analysis methodologies that are employed. Consequently, the most effective way to learn how to successfully use simulation is through learning how to apply it.

A major challenge in teaching applied simulation is the question of how to effectively blend and balance an understanding of fundamental principles and concepts with the practical side of building simulations. The intent of this book is to help bridge that gap and improve the effectiveness of simulation courses. Not all readers of this book will become simulation experts, but hopefully they will want to utilize the technology to help them or others make better decisions. Consequently the material takes the reader through three levels of users: Occasional, Intermediate, and Advanced.

Flexsim was chosen for use with this text because of its ease-of-use and rich functionality that allows users to focus on simulation concepts and methods. This is not intended to be a *Flexsim* manual as the *Flexsim* Help files and tutorials are more than adequate for that purpose.

The opening chapters focus on Occasional Users and provide a base for all user levels. The chapters establish the professional practice of applied simulation: the basics, economic justification, when simulation is needed, and a methodology for defining a simulation project. The intent is to demonstrate how simulation modeling and analysis is used to understand and resolve practical problems.

The next section in the book deals with Intermediate Users, those who desire to build simulations, but do so infrequently. These chapters focus on the basics of simulation software, statistics, equipment reliability, designing experiments, and model development. Simulation software details, if desired, are available in the appendices or from the *Flexsim* Help system.

Chapters for Advanced Users introduce topics such as writing custom logic, dealing with production schedules, and simulating fluid flow. Later chapters include a discussion of simulation software architecture along with examples of more advanced applications such as distributed simulation and agent-based simulation. Appendices cover *Flexsim* application details, application notes for each exercise, and specialized application topics.

Contents

Chapter 1 Simulation as a Tool for Understanding	1
Section 1-1 What is simulation?	1
Section 1-2 Working in a global, dynamic environment	2
Section 1-3 Simulation in everyday life	7
Section 1-4 Where is the money ?	10
Section 1-5 Simulation users	13
Chapter 2 Simulation Applications	21
Section 2-1 Modeling and simulation	21
Section 2-2 When to use simulation	25
Section 2-3 The simulation software marketplace	33
Section 2-4 Simulation and other tools	37
Chapter 3 Using Simulation to Solve Problems	45
Section 3-1 Using a simulation	45
Section 3-2 Simulations with random events	46
Section 3-3 Examples and exercises	47
Exercise 3-1 Coasting around	49
Exercise 3-2 Farm Pride	51
Exercise 3-3 Martian Transfer Station	53
Exercise 3-4 Slime Inc.	55
Exercise 3-5 Danson Electronics	57
Exercise 3-6 Grandma's Pie Emporium	58
Section 3-4 The next step	60
Chapter 4 Professional Practice of Simulation	63
Section 4-1 Confidence	63
Section 4-2 The Simulation Modeling and Analysis (SMA) Life Cycle	65
Section 4-3 The modeling and analysis process	66
Section 4-4 Roles in a SMA project	68
Section 4-5 Model-based decision-support systems	70
Section 4-6 Simulation modeling and analysis success factors	72
Chapter 5 Managing a Simulation Project	75
Section 5-1 The starting point	75
Section 5-2 Define the system and the reason	77

Section 5-3 Prepare a flow diagram	81
Section 5-4 Initial sizing of resources	87
Section 5-5 Building the simulation	88
Section 5-6 Results and analysis	90
Section 5-7 Example of a project template	92
Exercise 5-1 Fister's Frozen Foods	97
Chapter 6 Building Basic Simulation Models	105
Section 6-1 Simulation environment	106
Section 6-2 Simulation components	109
Section 6-3 Fixed resources	110
Section 6-4 Transporting items	112
Section 6-5 Editing objects	114
Section 6-6 Making connections	118
Section 6-7 Moving flowitems	120
Section 6-8 Creating simple learning models	122
Section 6-9 Single-run statistics	123
Exercise 6-1 Johnson Pharmaceutical	125
Exercise 6-2 Lucky Air	127
Chapter 7 Adding Model Logic and Managing Data	131
Section 7-1 Assigning attributes to objects	131
Section 7-2 Adding logic	132
Exercise 7-1 More Lucky Air	138
Section 7-3 Managing data tables	139
Exercise 7-2 Even more Lucky Air	140
Chapter 8 Managing Entities and Time Tables	143
Section 8-1 Grouping and ungrouping flowitems	143
Exercise 8-1 Hampton International	146
Section 8-2 Mobile resource objects (task executers)	149
Section 8-3 Displaying information on the screen	154
Section 8-4 Establishing time tables	155
Exercise 8-2 Steve's Stone Cutting	156
Exercise 8-3 The Crafty Frammer	158
Chapter 9 Modeling Randomness	163
Section 9-1 Data-driven probability distribution selection	163
Section 9-2 Selecting probability distributions in the absence of data	166
Section 9-3 Use of probability distributions in simulation	168


Section 9-4 Sampling from continuous random variates	169
Section 9-5 Sampling from discrete random variates	172
Section 9-6 Generating random numbers	173
Section 9-7 Putting it all together	175
Section 9-8 Using random samples to drive a simulation	176
Section 9-9 Reducing variability in the samples	182
Chapter 10 Analyzing Simulation Output	187
Section 10-1 Key tactical experimental design parameters	188
Section 10-2 Terminating versus non-terminating systems	192
Section 10-3 Determining the number of replications	195
Section 10-4 Creating experiments using Flexsim's Experimenter	196
Section 10-5 Comparing two alternatives	198
Section 10-6 Variance reduction through common random numbers	200
Section 10-8 Multiple comparisons of alternatives	201
Chapter 11 Including Reliability in a Simulation	205
Section 11-1 Performance	205
Section 11-2 Reliability	206
Section 11-3 Estimating MTBF and MTTR	208
Section 11-4 Simulating machine failures	210
Section 11-5 Setting MTBF and MTTR in a simulation	212
Section 11-6 Using personnel for repairs	215
Section 11-7 Surge	216
Exercise 11-1 Kegglers Brew	217
Exercise 11-2 Chairs for Tots	220
Chapter 12 Customizing Model Logic	227
Section 12-1 Hierarchical software architecture	227
Section 12-2 Understanding how objects work	231
Section 12-3 Scripting basics	233
Section 12-4 Creating custom logic	238
Section 12-5 Custom logic example	244
Exercise 12-1 Hilltop Steel Works	248
Chapter 13 Communicating Among Objects	251
Section 13-1 Communicating between objects	251
Section 13-2 Label Tables	256
Section 13-3 The recorder	257

Section 13-4 Customizing object placements	258
Section 13-5 Reusing custom objects	260
Exercise 13-1 Fister's Express	261
Exercise 13-2 Peoples Surgery Center	264
Chapter 14 Simulating Fluid Flow	269
Section 14-1 Basics of fluid-flow simulation	269
Section 14-2 Fluid objects	271
Section 14-3 Specifying Fluid Operations	273
Exercise 14-1 James Peanuts	276
Exercise 14-2 Western Grain	278
Chapter 15 Simulating Production Schedules	283
Section 15-1 Controlling production lines	283
Section 15-2 System controller	284
Section 15-3 Line controller	286
Exercise 15-1 Custom Shapes, Inc	289
Appendix	295
Appendix for Chapter 3	297
Section 1 General toolbars	297
Section 2 Viewing the simulation using a mouse	299
Appendix for Chapter 5	301
Section 1 Simulation Project Template	301
Section 2 Fister's Foods exercise	305
Appendix for Chapter 6	309
Section 1 Modifying the basic View settings	309
Section 2 Selecting and modifying object visuals	310
Section 3 Picklists	312
Section 4 Conveyor layout	313
Section 5 Exercise 6-1 Johnson Pharmaceutical	315
Section 6 Exercise 6-2 Lucky Air	316
Appendix for Chapter 7	323
Section 1 Putting labels on flowitems	323
Section 2 Modifying trigger logic with picklists	324
Section 3 Example 7-1 More Lucky Air	325
Section 4 Exercise 7-2 Even More Lucky Air	331

Appendix for Chapter 8	335
Section 1 Exercise 8-1 Hampton International	335
Section 2 Connecting and calling task executers	336
Section 3 Configuring timetables	340
Section 4 Exercise 8-2 Steve's Stone Cutting	342
Section 5 Exercise 8-3 The Crafty Frammer	343
Appendix for Chapter 9	345
Section 1 Notes for Using <i>ExpertFit</i>	345
Section 2 Statistical distributions in Flexsim	346
Section 3 Tables of normal and t distributions	351
Appendix for Chapter 10	355
Section 1 Setting up the Experimenter	355
Section 2 Defining scenarios	357
Section 3 Advanced experimenter options	358
Appendix for Chapter 11	359
Section 1 Exercise 11-1 Kegglers Brew	359
Section 2 Exercise 11-2 Chairs for Tots	360
Appendix for Chapter 12	363
Section 1 Commonly-used commands	363
Section 2 <i>Flexsim</i> software architecture	368
Section 3 Exercise 12-1 Hilltop Steel	371
Appendix for Chapter 13	375
Section 1 Recorder details	375
Section 2 Exercise 13-1 Fister's Express	376
Section 3 Exercise 13-2 Peoples Surgery Center	379
Appendix for Chapter 14	383
Section 1 Exercise 14-1 James Peanuts	383
Section 2 Exercise 14-2 Western Grain	385
Appendix for Chapter 15	387
Section 1 Custom Shapes Exercise	387
Appendix - Interacting with Other Applications	391
Section 1 Microsoft Excel import/export	391
Section 2 Importing AutoCAD drawings	392

Appendix - Advanced Techniques	395
Section 1 Task Sequences	395
Section 2 Creating AVI files to document the simulation	397
Section 3 Creating custom GUIs	398
Section 4 Kinematics	402
Section 5 Reports, statistics, and documentation	403
Appendix - Related Applications	407
Section 1 Health Care	407
Section 2 Container Terminal (CT)	408
Section 3 Express It 3D presentation software	409
About the Authors	411

Note: The following are samples taken from the book.

The  icon indicates that the text continues in the book.

Chapter 1 Simulation as a Tool for Understanding

This first chapter sets the foundation for studying simulation as an applied technology and analyzing and solving problems that exist in real systems. It provides the context for the book by defining what simulation is and the characteristics of the systems in which simulation is applied. These systems are widespread and diverse, including all areas of manufacturing and mining, as well as service operations such as healthcare, security, transportation, retail, and distribution, to name a few. The chapter introduces why simulation is used and what value it adds to all facets of decision making. It also describes the capabilities and skills of the three types of simulation users – Occasional, Intermediate, and Advanced. The book is organized into three main sections that address the needs of these three types of users.

Section 1-1 What is simulation?

As the title indicates, this book is about simulation with an emphasis on applied, real life situations. The primary focus will be on modeling and analysis. Before starting the book's journey the term *simulation*, as it is used in this text, should be defined.

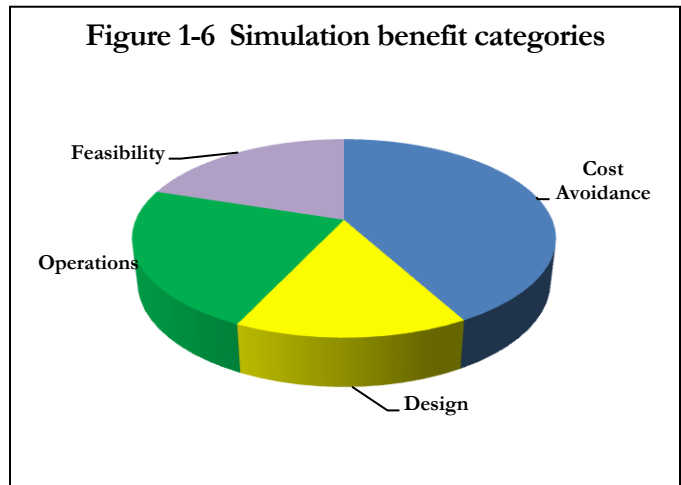
Generally, the verb *to simulate* means to mimic or imitate. This book takes a broader view of the verb as it relates to actual applications. Here, *to simulate* means to mimic or imitate through experimentation with a model (or representation) of some real system; however, simulation also involves more than just mimicking or experimenting. It involves performing such activities as defining, designing, and constructing a model or representation; defining the experiments to be conducted; collecting and analyzing data to drive the model; and analyzing and interpreting the results obtained from the experiments. Therefore *to simulate*, as referred to in this book, means to partake in a process that encompasses all of the above activities.



Section 1-4 Where is the money ?

Simulation is used for a reason—it has to provide a value-added return for the people and the financial resources it uses. Some of that payback is in hard cash while the rest is in soft, intangible benefits. Defined savings helps everyone—especially accountants—while the intangible benefits are appreciated by management. The examples used in this section are based on more than a thousand simulation projects over a wide variety of applications used in various business and service sectors.

The use of simulation can be justified based on experience



As shown in Figure 1.6, benefits of simulation normally fall into four categories: feasibility assessment, cost avoidance, detail design, and operations. Each category has both tangible and intangible benefits associated with it. As with most characterizations, there is always some overlap between the categories. The categories also line up with normal procedures for evaluating capital projects or any change to existing operations.

Using simulation for early decision making can result in significant savings

Feasibility Assessment:

In the feasibility phase of a project, new concepts are thought of and plans are made to execute them. It is the optimal time to make changes and identify possible problems since only a small amount of effort has been expended. Typical savings during this phase have ranged from 20 to 40% of the total cost incurred during this phase. Direct savings have accrued from reducing the time to obtain an accurate estimate of the cost and benefits of a proposed project. With simulation, the time to consider and prioritize alternative recommendations is also reduced. Time savings can be directly calculated from employee cost/hr and the value of a faster time to market from business projections.

.....

Chapter 2 Simulation Applications

Before learning how to build simulation models and analyze their results, it is important to understand what models are and how they can be used. This chapter investigates the many types of operational problems that simulation can, and has addressed. The application examples are certainly not exhaustive, but should serve to demonstrate the versatility and power of simulation. The chapter also provides a brief summary of the simulation software marketplace in order to put *Flexsim*, the software used in this book, in context. The chapter also discusses the role of simulation in conjunction with other types of applications found in most enterprises.

Section 2-1 Modeling and simulation

The role of simulation has been known for some time; however, it's only been since the early 1980s that simulation applications have reached a point where they are truly accessible to a wide range of users.

Modeling as a starting point

Simulation starts with a model. A model in this sense is defined as

a physical or mathematical description of an object or event.

A model usually represents a single point or action in time. A simulation then puts the model in motion. That is, they are dynamic—their states change over time. An operating scale model of a piece of equipment is called a physical simulation. Modern simulation applications are based on mathematical and logical models, but can look physical as well through the use of three-dimensional graphics.

Static modeling has its origin in ancient time

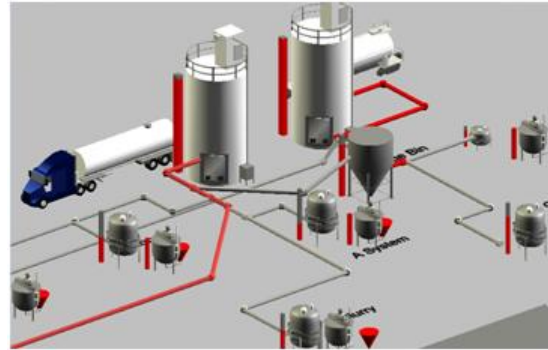


Issue 9: Improving raw material distribution within a facility.

Raw material receiving is often a neglected area but can be improved through simulation

The beginning or end of a system containing many operations is often a shared resource. A common example is the use of a shared raw material facility or shared warehouse area. While bottlenecks are known for the individual operations, the impact of the shared facilities is often neglected until it actually causes a loss in performance.

Distributing raw material to multiple plant sections

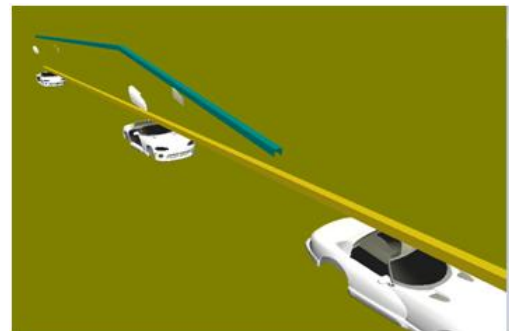


The speed of getting material from an incoming dock to its destination is often dependent on the routing and availability of transportation resources. The same is true of moving finished goods to their correct warehouse or loading dock destination. In medical simulations, shared resources of X-Ray or lab areas cause similar problems. Because of the interrelated nature of these operations, simulation is required.

Issue 10: Effectively coordinating a production line.

Coordination with other areas in a system is similar to the problems caused by shared facilities as discussed in Issue 9. In that case, however, each processing operation is still independent of the shared equipment. With coordination, two or more independent operations must interact at the exact same time. For example, in an automobile production line, components manufactured on independent lines must arrive at an assembly point at the same time in order to fill a particular order.

Automotive assembly lines require coordinated operations

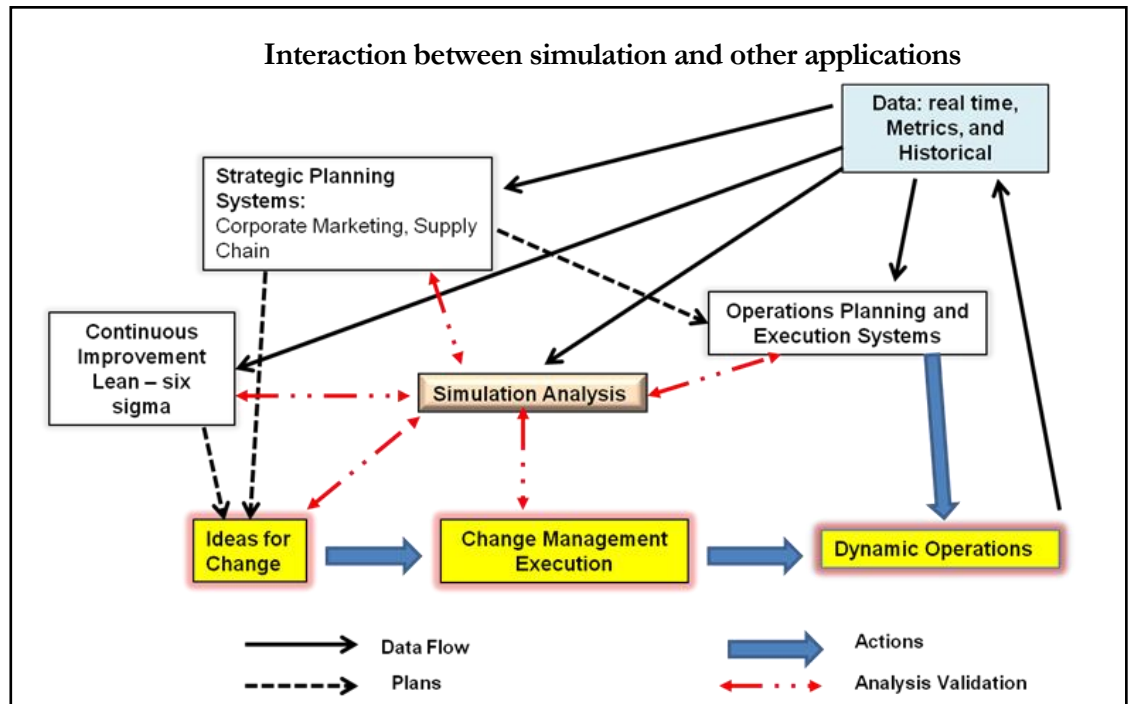


Section 2-4 Simulation and other tools

Each organization in a corporate environment usually has its own applications to help in carrying out their assigned functions.

Simulation is only one of a number of tools available for effectively studying dynamic systems. Other types of tools used extend from directing day-to-day operations to planning changes to meet new business opportunities. It is difficult, if not impossible, to find one tool that easily satisfies the needs of all users across such a wide range of activities; however, all of these tools share in the objective to improve system performance as they also share in the data and metrics that drive operations. It is therefore critical that the tools are able to share information across applications as well as organizational boundaries. Figure 2.19 shows the interactions between the various organizations and the tools that support the operations.

The Dynamic Operations block can be a production line, a plant, a service organization, or even a single operation. It is this operation that is the focus of the improvement efforts. Understanding its dynamic behavior and performance measures is the underlying goal. Two forces directly change the operating characteristics of the system. The first force is the day-to-day directives for operation.



⋮

Chapter 3 Using Simulation to Solve Problems

It is important for all stakeholders in a simulation to understand the power of simulation and how it can be used to solve problems. This chapter uses six diverse models to illustrate how simulation can be applied to improve system performance. All models employ a custom interface so that the users play the role of decision maker and Occasional User. The focus is on analysis and decision making and not on software and model building – that comes later in the book.

Section 3-1 Using a simulation

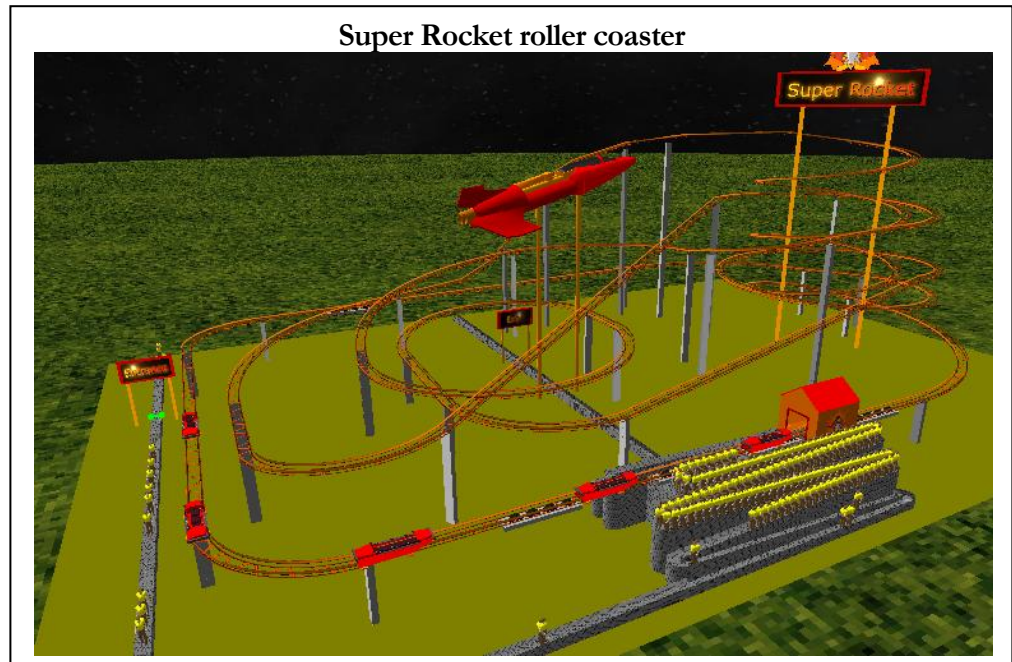
Simulation is an applied technology. There is little value to a simulation built without a reason or a problem to be solved; therefore, simulation must be experienced to be appreciated and understood. This chapter is an introduction to simulation applications through the eyes of an Occasional User.

An Occasional User, as defined by this book, is one who doesn't use simulation on a regular basis but appreciates the value of simulation. Usually this person's main responsibility is something other than simulation—a manager, lead engineer, team leader, etc. The Occasional User is not proficient in building simulation models but, with a little review, can make use of a simulation built by others, especially if the simulation is intuitive. It is assumed, however, that the user is familiar with some of the basic concepts of simulation.

The Occasional User knows enough about simulation to create a functional specification for others based on his or her knowledge of the system being simulated. As the “owner” of the problem or issue to be studied by the simulation, the Occasional User can detail the simulation requirements, define the scope, and establish the metrics for the analysis.



Exercise 3-1 Coasting Around



Background:

As the manager of the “Super Rocket” roller coaster ride at the *Grand Bay* amusement park, you are responsible for the day-to-day operation and financial performance of the coaster. Your parent company owns the coaster and leases the space from the park. The coaster at this park is only three years old and is the lowest in financial performance of all the parent company’s assets.

This year you, the manager, have to show improved financials for the ride or find another job. There are many trade-offs to consider. You are in competition with all the other rides at the park. You get paid by the park based on the number of riders served. Customers may love the coaster, but if they have to wait in line too long, they will walk away or not bother to ride again that day. Potential customers who see long lines will opt for other rides with shorter lines. Most choices to increase riders involve increased costs of equipment or staffing.

Your brother-in-law happens to be familiar with simulation methods and offered to help. You explain your problem and provide base data about the operation. You are appreciative of the results he gave you and decide to run the simulation yourself.

Problem statement

How can the profitability of the coaster be improved during the operating season?



Chapter 4 Professional Practice of Simulation

It is of paramount importance to have the results and recommendations derived from a simulation model accepted and implemented. After all, the power of simulation and all of the effort expended to build models and conduct analyses is lost if it is not used. The most technically elegant model is of no value if it does not support decision making in a timely manner. This chapter addresses the non-technical aspects of simulation modeling and analysis, including the importance of user confidence, understanding that simulations have a life cycle and involve a definitive process, and the many roles that must be fulfilled in order for simulation to be applied successfully. This chapter also discusses model-based decision-support systems and concludes with a set of recommended success factors that are critical in simulation projects.

**Credibility,
built on
confidence
is key to
obtaining
acceptance
for the use
of
simulation**

Section 4-1 Confidence

When a person has a serious medical condition and a new technology is available, confidence is a major factor in choosing whether or not to use the new approach. Not only confidence in the technology itself, but also in the person using the technology and in the process by which the technology is applied. Of course, a person who is desperate for help may choose the new procedure even without total confidence. Decisions to use simulation can be very similar. Simulation is a tool for analyzing and solving problems associated with operations systems. As with the medical example, it is a technology that may be new and unfamiliar to those seeking help in solving a problem. Although an intense desire to find an answer to a problem may outweigh other considerations, confidence is critical to the successful use of simulation and to the professional success of the person using it.



Section 4-4 Roles in a SMA Project

Too often the roles of individuals in a SMA project are considered solely within the nomenclature associated with the project activities described above. As a result, the only roles that are explicitly addressed are those that deal with model construction, and maybe analysis. However, there are many roles or actors who provide valuable contributions to a successful SMA project. Of course, some individuals may fulfill a variety of roles. Those roles are defined below.

Simulation projects involved many tasks that may easily overlooked



- **Developer:** An obvious role involved in developing and constructing the simulation model and/or model support software, such as database connectivity, user interfaces, decision support systems, etc.



- **Designer:** A closely related role to that of developer, but focusing on establishing the model and/or decision-support-system architecture.



- **Analyst:** Designs (e.g., defines what variables need to be changed in each scenario and their value, determines run length, warm-up period, number of replications, etc.), conducts, and summarizes experiments.



- **Researcher:** Early in any project, this individual conducts research to see if similar issues have already been addressed elsewhere and what can be learned from previous work.



- **Investigator:** Since simulation depends on a clear understanding of the system being modeled, it is imperative that processes are well defined and data are available; this requires a significant amount of probing and “digging.”



Chapter 5 Managing a Simulation Project

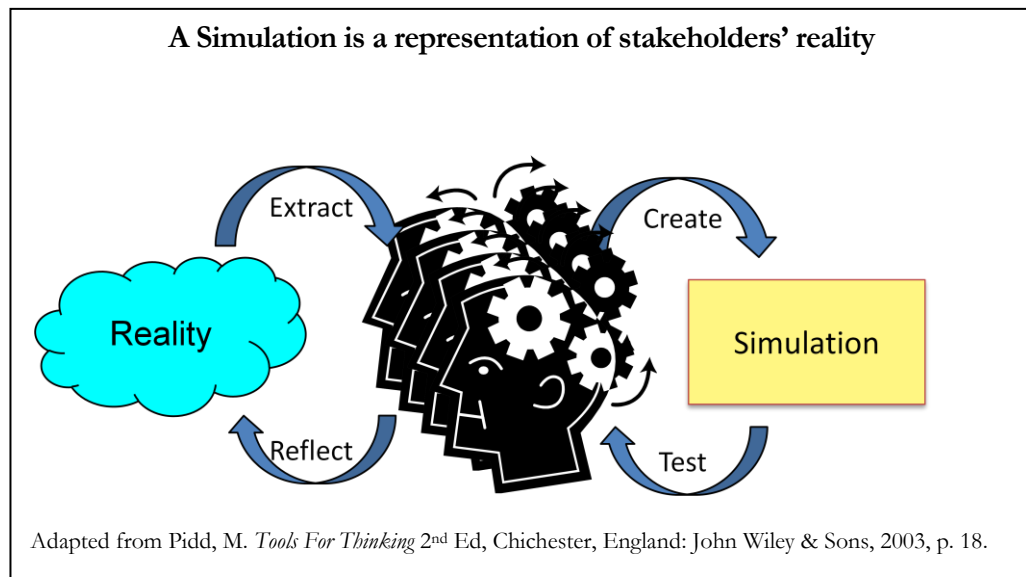
Chapter

5

The previous chapter provided a general outline of how the SMA process is used for analyzing and solving problems with simulation. It highlighted the people and organizational issues involved with the successful use of simulation. This chapter focuses on the technical details of carrying out the SMA process and provides tools that can be used in the process.

Section 5-1 The starting point

An organization that needs a simulation for analysis must follow a series of steps in order for the simulation to be successful. Building a simulation involves translating reality into a time-based model. As individuals look at the actual system, they develop their own conceptual model of how the system operates. Each person may see the system slightly differently or notice different characteristics—all of which may be important. Consequently, a methodology is needed to aid in communications and general understanding.

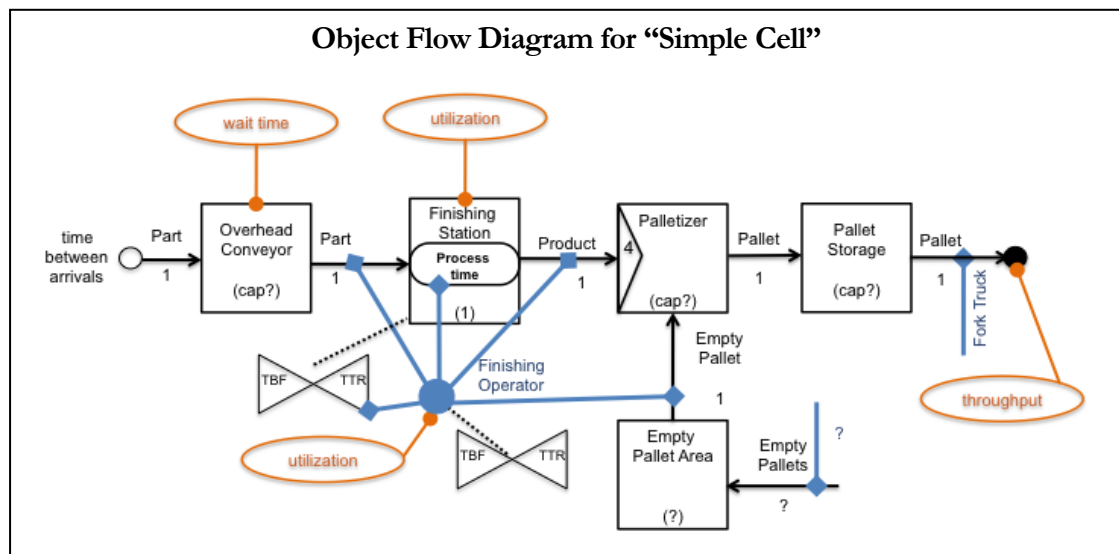


Section 5-3 Prepare a flow diagram

The best way to document the material obtained for the functional specification is with a flow diagram of the system. Such a diagram also becomes a useful communication tool when discussing the project.

Most operating systems in manufacturing, service, logistics, or other areas can often be described at their simplest level with a diagram showing each step as a block. There are many ways to represent processes diagrammatically; for example, detailed CAD (Computer Aided Design) diagrams and P&IDs (Process and Instrument Diagram). The P&IDs provide very detailed descriptions (usually in 2 dimensions) of all equipment, piping, electrical, and other connections.

Another type of diagram, the value stream map, is often useful in modeling since it contains considerable process data that have already been investigated, discussed, and compiled. An example value stream map is provided in Figure 5.3. This type of diagram, however, is not sufficient by itself for defining a simulation model since it typically lacks the detailed logic needed in most simulation models (e.g., rules for making material transfer decisions or detailed sequence of operations).



.....

Chapter 6 Building Basic Simulation Models

Chapter

6

The Intermediate User uses simulation as part of his or her job description, but it is not usually the main focus. Typical job titles of the Intermediate User include operations analyst or industrial, process, packaging, or manufacturing engineer. Individuals engaged in manufacturing research and development may also be Intermediate Users. Typically, the Intermediate User will be involved with simulation five or six times per year. Intermediate Users can build relatively straightforward simulation models using drop-down menus or wizards. When more detailed simulations are required, the Intermediate User can call on technical support or third-party consultants.

Using simulation as a basic tool can be accomplished with the use of pre-built logic and a minimal amount of training

This chapter introduces the basic structure and components found in most simulation software modeling and analysis packages. It includes a discussion of the modeling environment, introduces the basic structure and functionality of modeling objects, defines relationships among objects, describes how to move items through a model, and discusses obtaining output statistics from a single run of a model. This chapter will illustrate the object functionality as used in the *Flexsim* application. The remainder of the book builds the reader's capability to create and analyze simulation models through the *Flexsim* software. While *Flexsim* is used in this explanation, the functionality is similar to other applications, but is implemented differently.

This chapter, and those that follow, contains exercises that will develop those simulation skills used by Intermediate and Advanced Users. These skills, which are needed to build simulations, are discussed in general terms in the main chapters, while specific details of the *Flexsim* implementation are contained in the Appendix.

The discussions in this chapter are meant to be a quick-start introduction to simulation modeling. Some of the topics will be explored in greater detail in later chapters. Consult the Appendix as well as the Users' Manual in the Help section of *Flexsim* for further information about using the software.



Exercise 6-2 Lucky Air

Background

Lucky Air, a start-up airline, is committed to providing a shuttle service between the Orange County airport and the Las Vegas airport. With their fleet of regional jets, the company feels that the time is right to bring casino patrons for quick daily visits to Las Vegas. Their promise is to fly as long as there are people who want to travel. If a scheduled plane is full, another will be brought out; their motto is *Always a Winner*. They expect an increase in business as people try to find money from the slot machines during the downturn in the economy.



The owner wants to set up operations as quickly as possible so he decides to operate their check-in counter with three ticket agents: one for passengers with e-tickets, another for passengers with paper tickets, and a third for passengers purchasing a ticket. As the only engineer in the new airline, you don't think the level of service will be good and the agents will be working inefficiently. Since you don't want to verbally confront the owner, you decide on showing what might happen with a simulation.

Problem statement

You believe the proposed operation will be inefficient and you want to illustrate the system's behavior to the owner.

Application Discussion:

- How could the following objects be used to represent the waiting line?
 - a queue – what options would make it look like a line?
 - a conveyor
 - a flow node
- Could a processor be used as a conveyor – how? Under what assumptions?
- What are the differences in the four constructs identified above for getting people to the ticket window? Does the choice significantly impact the results?



Chapter 7 Adding Model Logic and Managing Data

The last chapter provided the basics for building discrete-event simulation models. It defined the functionality of some common objects, as well as some of the main properties of those objects. It also illustrated how to bring objects into the model and connect them so that the model represents a system of interest. This chapter explores some of the object properties in more detail and presents means to add logic to a model to enhance its representation of a system's behavior.

In the last chapter, information about an object's behavior was straightforward since each object was designed from the start to work in a particular way (e.g., a processor invoking a planned delay to represent service time). But what if the object's behavior changes based on the circumstances at the time, or if it changes when the value of certain parameters reaches a particular level? The person building the simulation model must decide where to best store object information so that it can be used appropriately to modify behavior or logic.

The choices and methods vary by simulation application, although the basic functionality is usually similar. In some cases the application itself may limit the choice; and in other cases, many options may be possible, and the decision is left to model developer. At times the choice can be based on preference or design style, while at other times the choice is dictated by the system being simulated.

Section 7-1 Assigning attributes to objects

Most applications allow attribute information to be maintained on the objects themselves. This construct in *Flexsim* is called a Label. Attributes or labels, provide a means to associate specific properties with an object or to maintain information about an object. Labels are created on objects by opening the Labels tab on the properties window of an object.



There are various ways to add data to objects or flow items.

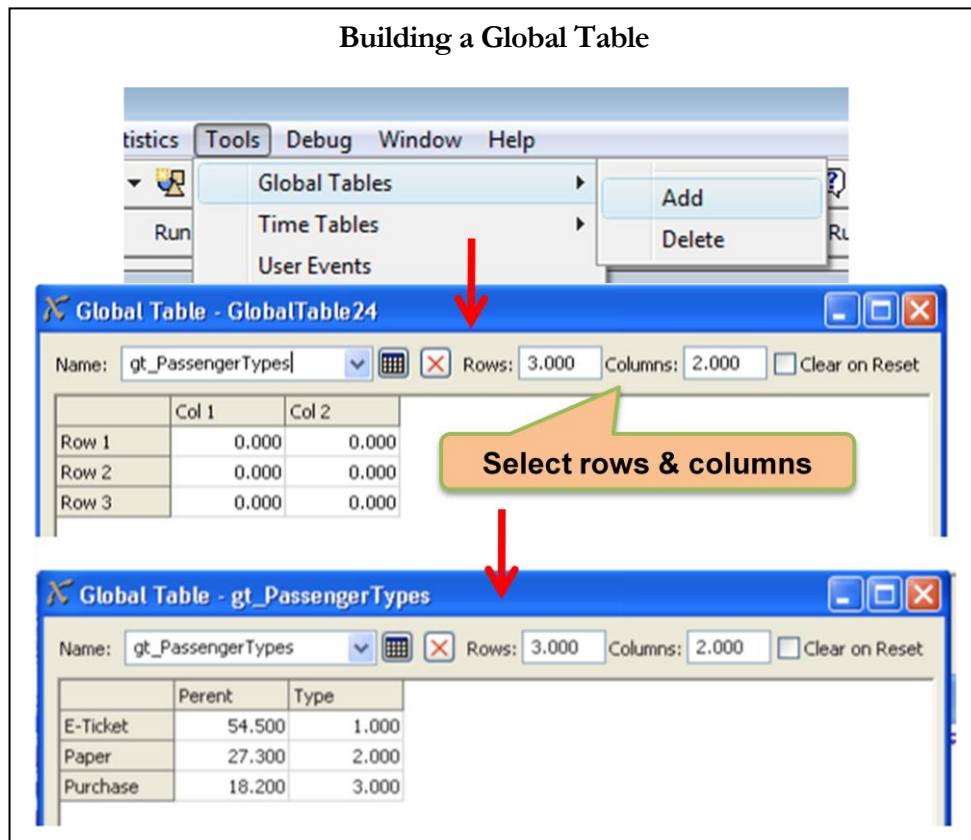
Section 7-3 Managing data tables

Since simulation models are data-driven, managing the model's data is an important issue. One important practice is to remove the data from the internal logic of the model so that can be easily changed for sensitivity analyses and manipulation by less experienced users. In *Flexsim*, one convenient way to store data, both input and output, is via a Global Table object. These table objects can be read from, and written to, during the execution of the simulation model.

Since global tables are needed in the next exercise, one will be setup here as an example. The context on how the table is used in the example is explained in the Appendix.

Global tables provide a location for data that is easily accessible throughout the simulation.

Global tables are added and edited through the Global Tables option on the Tools menu, which is part of the main Menu bar. Choosing the Add option creates a simple single-row, single-column table that can be modified by specifying the number of rows and columns in the appropriate boxes. The table for the next exercise is shown in Figure 7.6 with the basic table structure shown first, followed by the completed table.



Chapter 8 Managing Entities and Time Tables

This chapter extends the Intermediate User's modeling capability in several key areas. In many situations items need to be grouped and ungrouped as they flow through a model - this is discussed in the first section. Mobile resources are one of the main classes of objects in any simulation software. In Flexsim mobile resources are referred to as task executers; these are especially versatile and powerful objects. The second section of the chapter describes the various types of task executers including their basic operation and the main properties that drive their behavior. In order to enhance visualization additional information oftentimes needs to be displayed on the simulation surface; this information may include statistics, state status, textual descriptions, etc. Section 3 describes how this information can be displayed in Flexsim. The final section describes the use of time tables to control object states. The chapter uses several comprehensive examples to illustrate these concepts.

Section 8-1 Grouping and ungrouping flowitems

Grouping and ungrouping items are common events in a simulation

There are many situations in operations systems in which the items flowing through the system are grouped or ungrouped. Examples include packing items in a tote for handling or in a box for shipping, unpacking parts from suppliers for assembly, unpacking totes after movement, making multiple copies of an item and distributing them in the system, etc. In *Flexsim*, these operations are handled by the fixed resources known as *combiners* and *separators*. Combiners and separators share many of the standard attributes of other fixed resources, especially processors, but have some specialized features of their own



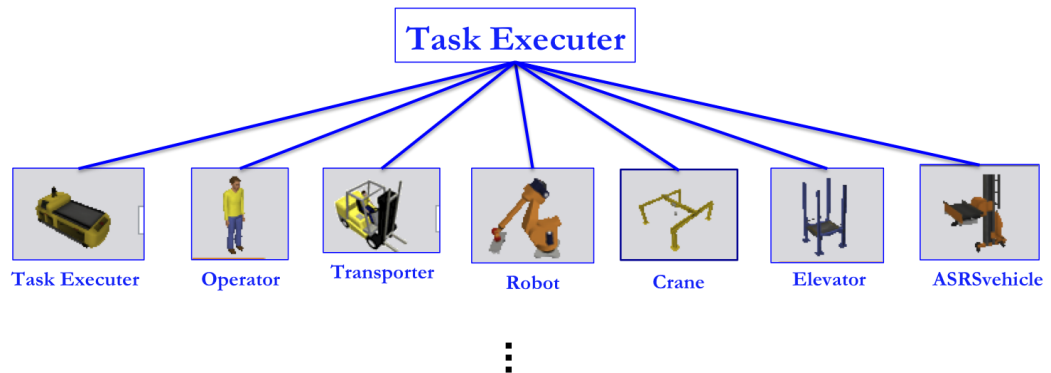
Section 8-2 Mobile resource objects (task executers)

Mobile resources can both carry items and move over the simulation surface

In addition to the objects previously described to move flowitems, simulations often require mobile resources to move or carry items. These resources can include forklifts, guided vehicles (AGVs), or operators. Simulations may also require mobile resources to carry out certain tasks such as setting up or repairing a piece of equipment. The mobile resource objects, also referred to as task executers in Flexsim, can perform various functions:

- Physically move over the simulation surface
- Carry or transport flowitems between objects
- Execute a series of tasks
- Be assigned to an object in association with a particular operation (e.g., processing setup, processing, maintenance)
- Distribute tasks to other task executers
- Follow a path created by network nodes

As shown in Figure 8.6, *Flexsim* offers a variety of types of task executers in the standard Discrete Objects library. In terms of their basic functionality, the objects are very similar. Note that there is a task executer object within the general class of task executers



Section 8-4 Establishing time tables

Time tables can be used to control an object during specific times in the simulation.

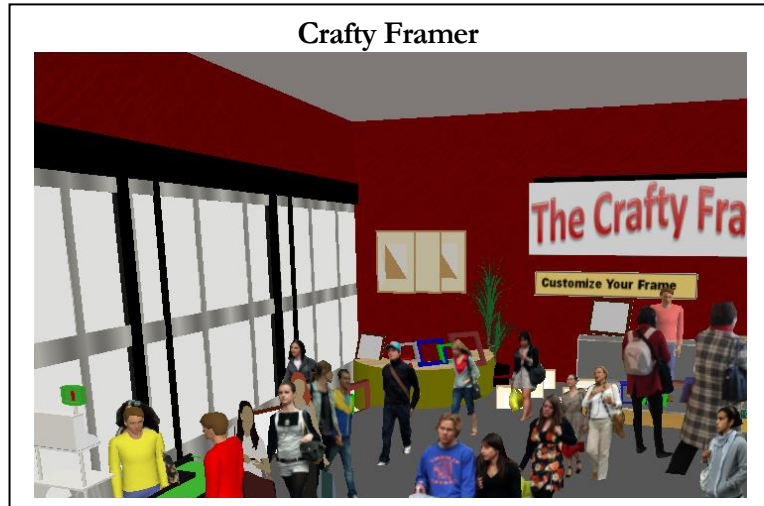
During the course of a simulation, many objects may start or stop, depending on the time of day. For example, workers may not be available during break times, or equipment may only be used on certain days. A common occurrence is stopping or starting equipment based on a particular work schedule. Often, custom programming is required to create and manage work schedules. In *Flexsim*, such time tables are established by building a time table in the Tools tab located on the main menu bar at the top of the screen.



Exercise 8-3 The Crafty Framers

Background

Your cousin has a great idea to sell customized picture frames and wants to get set up in time for the Christmas holiday season. The store is in a small location where customers can choose picture frames in all sizes and styles; however, the most profit will be made if customers choose to have the frame customized with specific decorations.



The customization process is straightforward and doesn't take much time, and the higher price can mean a greater profit margin. Since you mentioned having some knowledge of simulation, your cousin has asked you to simulate the shop operation and help decide how to best utilize the workers in the store. He currently has one checkout person and one custom framer but will hire another worker for the holiday rush.

Problem statement

Simulate the frame shop during a ten-hour period in order to help decide how to best utilize the three workers in the store.

Operating Data:

The frame shop operates from 9 a.m. to 7 p.m. At 7 p.m. the front door to the store locks. Customers in the store are serviced and the store is cleaned until 9 p.m. The procedure is repeated every day.

As shown in Figure 8.12, the store has two service counters—one used for customizing frames and one used as a check out area. The three workers can handle either the check out or customizing station. During the holiday shopping period, customers are assumed to arrive at the shop at an average rate of 56 per hour; assume the times between arrivals are exponentially distributed. There are three types of customers who enter at different rates:

There are three types of customers who enter at different rates:



Chapter 9 Modeling Randomness

Simulating real systems means simulating randomness.

Recall that a model is a representation of a system. The model is built to understand the behavior of the system and to help assess the consequences of actions on, or by, that system. A key aspect of model development is capturing the essence of the underlying structure and logic that exists in the system. Since discrete-event simulation models are stochastic (some of the inputs are probabilistic), another key aspect of model development is discerning the nature of the probability distributions used in the model. Many of the values used in a discrete-event simulation—times between arrivals, service times, quality indicators, time between failures, etc.—are obtained by taking random samples from probability distributions.

This chapter focuses on methodologies that are used to help decide what probability distributions to use in a model. This decision is either based on sample data obtained from the system, or it is assumed in the absence of data; both of these approaches are discussed in this chapter. The chapter also discusses how simulation software generates random samples from probability distributions—this includes discussing both generating random variates and generating random numbers. The chapter also discusses how random samples drive events and the simulation itself.

Section 9-1 Data-driven probability distribution selection

Where data is available, a statistical distribution to represent that data may be found.

Oftentimes distributions that are used in simulation models are based on data obtained from the system being modeled. This involves taking samples from the random processes (e.g., service times or times between arrivals), “fitting” that data to standard or theoretical probability distributions (e.g., normal, exponential, lognormal, or Weibull), and inferring which distribution best represents the population in the real system. If the sample data does not fit a standard distribution well, then an empirical distribution is used; that is, one that closely approximates the probability distribution of the sample data. In both cases, a probability distribution is selected based on data sampled from the system being modeled.



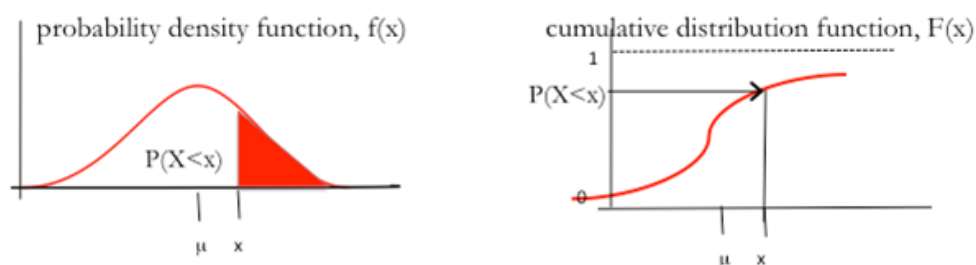
Section 9-4 Sampling from continuous random variates

Once probability distributions are specified in a simulation model, the software must repeatedly sample from those distributions as the model runs. In order to invoke randomness in a simulation, the model developer only has to specify the probability distributions and their parameters. It is, however, useful to know how the sampling occurs within the software. This section describes the process used by most simulation software to sample from continuous probability distributions; the next section describes the process for sampling from discrete probability distributions. In general, consider $f(x)$ to be a continuous probability density function, and $F(x)$ is the cumulative distribution function where $0 \leq F(x) \leq 1$. In general, $F(x)$ is defined as:

In the simulation, samples are taken from specified distributions.

$$F(x) = \int_{\text{lower limit}}^x f(w) dw .$$

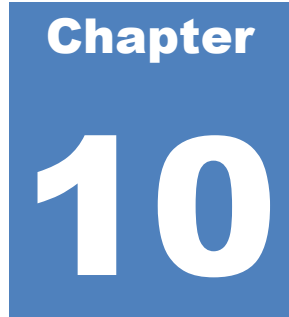
As shown in Figure 9.7, $F(x)$ represents the probability that the random variable X will be less than the specified value x ; this is also the area under the $f(x)$ curve from the lower bound of $f(x)$ to x .



Also, consider u_i to be a random number uniformly distributed between 0 and 1. Since u_i and $F(x)$ have the same domain (they can take on values between 0 and 1), u_i is set equal to $F(x)$, and the equation is used to solve for x in terms of u_i ; that is, . This approach is referred to as the inverse transformation method. Two examples are provided to illustrate how the inverse transformation works—sampling from the uniform and exponential distributions.

⋮

Chapter 10 Analyzing Simulation Output



A simulation model is developed in order to generate output for analysis. Since models are representations of real systems, they are used to understand the behavior of those systems. Some of the inputs to a model are values for decision variables (i.e., values that are controlled by the decision maker). Output from a model provides information on the consequences of setting the decision variables in a certain manner. A model is like a laboratory, and as such, experiments are designed and run using models.

This chapter begins with a discussion of the definition of key tactical experimental design parameters. It is followed by sections covering

- two types of systems, terminating and non-terminating, and their impact on setting the design parameters;
- a means for determining how many replications of a model are needed;
- Flexsim’s Experimenter, a very powerful means for conducting experiments with simulation models;
- how to compare output obtained from two simulated alternatives;
- the variance reduction technique common random numbers (CRNs);
- how to perform multiple comparisons between more than two alternatives.

Careful statistical analysis is paramount when using output from a discrete-event simulation model. In order to draw conclusions and make valid inferences, simulation models must be replicated; that is, the models must be run multiple times. This involves independent random samples provided from the various statistical distributions that can be used in a model. The information gathered from the multiple runs is combined to provide a basis for conclusions and inference. For example, confidence intervals are used to estimate the means of output or response variables which provide the foundation for inference and decision making. Methodologies for making inferences from simulation models are discussed later in the chapter, following a discussion of some important tactical experimentation variables that must be set in order for the model to generate the data needed for analysis

There are a number of parameters that have to be set before experimentation can begin



Section 10-2 Terminating versus non-terminating systems

The importance of experimentation parameters and the means to set them as described above depend on the type of system being modeled—terminating or non-terminating.



Section 10-3 Determining the number of replications

As discussed earlier, in order to avoid making inferences on a single, possibly extreme value, a model should be replicated a number of times and the values of the performance measures from each replication averaged. The general intent is to conduct experiments with the simulation model and estimate the population performance measures of interest, such as the mean. As shown in the formula below, the expected value of the population mean, μ , is estimated by the mean of the sample means that are obtained from each replication. Similarly, a measure of the dispersion in the estimates is given by the variance in the sample means.



Section 10-4 Creating experiments

Flexsim provides a very powerful analysis tool: the Experimenter. While the details for using the Experimenter are provided in the Appendix, the key interfaces are shown in Figure 10.8. The Experimenter requires three types of inputs that need to be specified whenever experiments are designed



Section 10-5 Comparing two alternatives

There are several approaches outlined in this section for comparing $k > 2$ alternatives. More information on these approaches can be found in Law, A. *Simulation Modeling and Analysis*, New York: McGraw-Hill, 2007 or earlier editions by Law & Kelton, e.g. 2000.



The appropriate number of replications may not be obvious

Automating the experimental process can speed simulation analysis

Chapter 11 Including Reliability in a Simulation

As systems become more complex, additional functionality is often required for a more accurate representation of operations. Reliability has a significant impact on the performance metrics for a system. It also plays an important role in determining the location and size of any material accumulation points, which is critical for establishing an efficient production system.

Section 11-1 Performance

Analyzing a system's performance in the face of random events is often the goal of a simulation. *Performance* can be a confusing term as the parameters that define it may change based on historical practice or other factors. The following are commonly used measures of performance and are often used interchangeably.

Performance measures can take many forms.

- *Maximum rate*: Best possible output rate based on design
- *Ideal rate*: Maximum output rate based on theory
- *Target rate*: Output rate that is considered consistently achievable
- *Planning rate*: Output rate used for planning and scheduling purposes
- *Efficiency*: Actual output rate measured against some specified metric (usually target output rate)
- *Performance*: Some metric indicating how well a system is operating
 - *System Performance*: Based on an entire operating line
 - *Unit Performance*: Based on a single piece of equipment
 - *Plant Performance*: Based on multiple systems



Section 11-4 *Simulating machine failures*

The level of detail used to simulate a system should be consistent with the reliability data available.

A single machine can fail in many ways depending on its activity

In some situations, a simulation may need specific information for individual pieces of equipment. For example, a case packer machine will have separate failure modes for different steps or processes, such as

- collecting materials to pack;
- a case building section;
- the filling process;
- a case closure section;
- an output transport section.

Additionally, machines may have different types of failure modes: short-term (jams), intermediate-term (in-feed or out-feed conveyor problems), and long-term (belt failure). These multiple failure modes are called competing failures. Each competing failure will have its own statistical distribution and may have its own failure clock.

Section 11-6 *Using personnel for repairs*

When dealing with downtimes, it is often necessary to simulate both the downtime itself and the response time of someone coming to perform the repair. To designate that another resource must be present to make the repair, choose the drop-down menu choice in the Down Function that will call an operator, as shown in Figure 11.9. All items in blue can be modified.

Personnel can be called to service a machine

Section 11-7 *Surge*

Surges are areas where material is stored while awaiting processing. Common surges include a raw material holding area or a warehouse where finished products wait to be shipped. A general principal of lean manufacturing is to minimized the use of surges as there are no value-added functions being performed. However, there are places where surges are useful

Providing for some level of surge can be a means of protecting equipment from downtimes

Chapter 12 Customizing Model Logic

While the Intermediate User can study a wide variety of dynamic systems using simulation, there will still be a need to simulate actions that cannot be described using the preset logic. The Advanced User has enough understanding of the simulation software to create custom logic and use advanced features to analyze more complex situations. Depending on the ease of accomplishing these tasks, the Advanced User may not need to be a simulation or programming professional.

Advanced users can handle more complex applications but may not necessarily be an expert in the simulation language.

Most Advanced Users will have either a definite interest in simulating systems or will carry out simulations on a relatively frequent basis. An Advanced User normally will also be knowledgeable of the subject matter that makes up the simulation problem. While this section develops an extended knowledge of the simulation software, it does not completely describe all of the advanced functionality available in the application.

The primary background required for the Advanced User is knowledge of the underlying software structure and the basic programming commands for building custom logic. The Advanced User also becomes more proficient in dealing with information that is exchanged between a simulation and external programs. Additional skills include utilizing the charting and reporting functionality of the software, enhancing the visualization of the model, etc. This section provides an overview of the software architecture and its scripting commands. More information is provided in the *Flexsim* Users' Manual, available through the Help menu.

Section 12-1 Hierarchical software architecture

The *Flexsim* software is unique in a number of ways from the other simulation applications discussed in Chapter 2. The most important is that it has an object-oriented software structure. Object-oriented programming was developed to improve functionality and maintain software quality as computer hardware and software became increasingly complex.



The tree is a visualization tool for the software and its hierarchical structure

The use of object-oriented techniques in the development of *Flexsim* means that it is hierarchical and data oriented in nature and consequently can be visualized by a tree structure used for databases. It's called a tree even though the visualization is generally upside down compared to an actual tree, with the "root" at the top and "leaves" at the bottom. A structured representation of a simple simulation containing four elements is shown as a tree in Figure 12.1. A tree structure is conceptual, and appears in several forms. Common examples of tree representations are the folder/file graphical interfaces found on computer systems, e.g. *Windows Explorer*. In *Flexsim*, the basic elements in its tree view are called nodes.



Section 12-3 Scripting basics

Scripting is the means to change logic and create custom, advanced functions

The picklist options, introduced for the Intermediate User, provide a wide range of commands to control the simulation without writing any code; however, if a simulation is intended to examine truly unique strategies that provide a definitive advantage, then writing some level of unique, custom code is to be expected. *Flexsim* uses a scripting language called *Flexscript* to create custom logic.

Flexscript is modeled around the syntax of the popular programming languages C/C++ and follows most of the same conventions found in those languages. Anyone familiar with C/C++ can quickly begin writing custom modeling code. Some basic rules and guidelines for using *Flexscript* include the following



***Flexscript*—commands**

***Flexscript* commands are custom, high level instructions that carry out common software operations needed for simulation.**

Flexscript contains a large number of commands that provide easy access for referencing or getting information from particular nodes in the tree structure. They are also used to perform other functions, such as interacting with *Excel*. The command choices under the Help menu provide definitions of all commands as well as examples of their use. Commands are arranged alphabetically and by primary function



Referencing objects

The tree structure of *Flexsim* gives objects hierarchical relationships to each other. These relationships allow for referencing objects quickly and easily using commands and key words. Custom logic usually means interacting with other objects for example, changing operations in one section of a simulation when a downstream section experiences a particular problem



Chapter 13

Communicating Among Objects and Enhancing Model Use

Most simulation languages have a capability to send information from one object to another in order to facilitate building complex logic. They also provide for various means of visually recording variables as the simulation progresses. In *Flexsim*, the communication is accomplished through messages from any one point in the model to another. *Flexsim* also provides a recorder object for monitoring variables. Most importantly, when custom, complex logic is built for objects, those objects can be saved to a user library for future use.

Section 13-1 Communicating between objects

Communicating between objects with messages can be a convenient and efficient way of sending information to coordinate and control the simulation. A message can be sent from any trigger on any object in the model to any other object in the model at any time during the simulation. The OnMessage trigger (located in the Triggers tab) on each object executes when it receives a message

Sent messages are received in the OnMessage trigger.



Section 13-5 Reusing custom objects

The Advanced User will often create objects with new images attached to them or include custom logic on the object, such as the machine seen earlier that had its own built in startup cycle. Such objects may be desired in other simulations, and significant time can be saved if custom work can be reused. In *Flexsim*, user libraries can be created and used to share objects.

Objects can be saved in custom libraries for future use.

Clicking on the New Library icon at the top of the Library tool bar opens a blank library. An object can be added to the library by first right clicking on the object and selecting Edit. At the bottom of the secondary window that opens is an option to add the object to the library.



Exercise 13-2 Peoples Surgery Center

Background

Given the changes in medical care being directed by the government, Carson Memorial Hospital has decided to open a surgery center for common out-patient surgical procedures. It is felt that such a center could be operated more efficiently and cost effectively. The hospital was originally designed for major procedures and longer-term care.

Before setting up the new facility, the hospital administration wants to have a simulation developed to check on the staffing levels that are needed and the amount of time patients would spend in the facility—especially in the admission and lab areas.

Problem statement

Provide a simulation to help establish operating conditions for the admissions and lab areas of the surgery center.



Modeling and analysis issues

- How can an entire system be modeled in sections to test it before putting it all together?
- What are different ways to bring the patients into the system? How can you regulate the times the center allows patients to enter?
- How can you identify the patient type and flag the type 2 patients who have come back from their lab work?
- What queue options will prioritize how patients leave the main waiting area?
- What logic is needed to send the patients to the right place?



Chapter 14 Simulating Fluid Flow

The majority of simulation application packages on the market are based on discrete-event simulation; however, interest in simulation has grown to include operations involving the continuous flow of material such as liquids, powders, and high speed bottling. Simulating both a processing and packaging step in a production line involves the transition between continuous and discrete events. A few simulation packages only simulate fluid flow, just as some only simulate discrete events. Others, like *Flexsim*, are capable of simulating both fluid and discrete systems in the same simulation model.

Section 14-1 Basics of fluid-flow simulation

Fluid Concepts

A material flow is considered to be a fluid when its moving material cannot be identified by individual objects but appears as a continuous body. Fluid, however, doesn't necessarily mean liquids. Liquids, gases, and small particles such as rice, cereal, and sand are obvious examples of fluid flow. A less obvious example of a fluid is a high-speed bottling line (> 1600 bottles per minute), it can be thought of as either a fluid or a discrete material flow. A fluid (or continuous) flow is usually described in terms of units per time, such as gallons per minute.

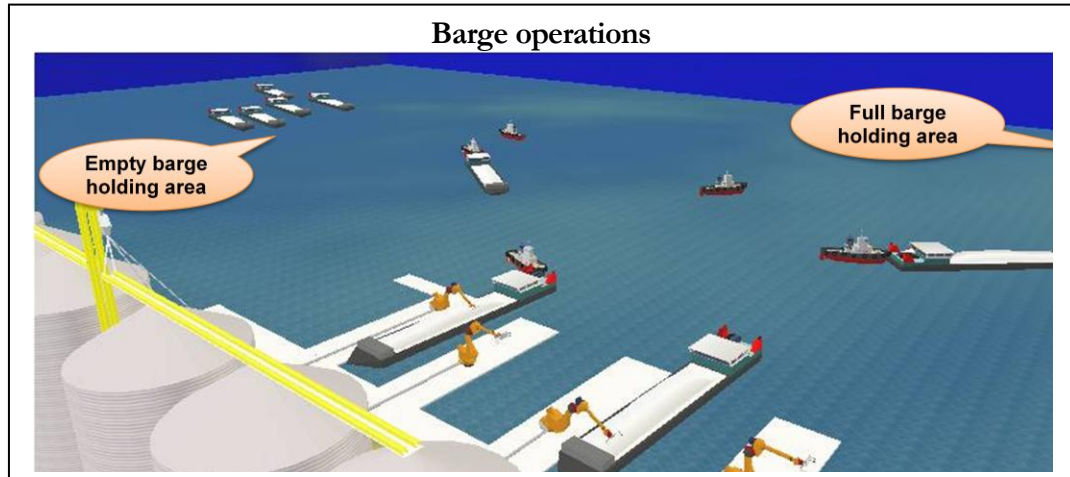
Being able to simulate fluid flow is important to industries that have both continuous and discrete manufacturing processes. The manufacturing involves a process area where fluid-type materials are produced as well as a discrete area such as a packing line. Discrete material flow has specific events associated with it, such as entry into a machine or exit from a queue. These events are placed into an events list for processing in time order by the discrete-event simulator. Fluid flow, however, moves continuously.



Exercise 14-2 Western Grain

Background

The Western Grain Cooperative is planning a new grain loading facility to transport grain from their main silos to barges on the Mississippi River. Grain is loaded onto barges at the facility's four docks. Before they go ahead with the plans, the members of the board want to be assured that the concept will work and that grain will not be left in the main facility. They want to be sure that they can ship between 55,000 and 60,000 tons of grain per day.



Problem statement

Determine the feasibility of the loading operation including the number of tugs that are required.



Modeling and Analysis Issues:

- What time unit should be the basis of the simulation?
- What level of detail will be used? What assumptions are made?
- What variables should be used and where should data be kept?
- Should fluid be used to represent the grain?
- What can represent a barge? Does it actually have to be “loaded”?
- How will the barge arrival schedule be simulated?
- How will the barges get to and from the docks?
- How will the travel time be simulated?
 - What are the performance metrics



Chapter 15 Simulating Production Schedules

In today's economy, few manufacturing operations constantly produce the same product. The drive toward flexible systems means that a single production line has the capability to produce a number of products. The order in which products are produced can greatly affect the operating performance of the line. This is especially true if variables such as the setup time between products, production rate, and machine reliability vary by product.

The production schedule is often dictated by a marketing or business unit based on actual or anticipated demands. Even if the production schedule is generated by an enterprise-level optimization application, the schedule may not necessarily be optimal for the local production line.. Simulation is used to validate the impact of a production schedule on specific manufacturing operations. *Flexsim* has two objects that were specifically developed to simplify the simulation of production schedule: the system controller and the line controller.

Section 15-1 Controlling production lines

Often the scope of a manufacturing simulation will include one or more production lines. The productions lines are controlled by a schedule which indicates what products are to be produced, what equipment will be used, and in what order the production should take place. Simulating such an environment would normally require a large amount of custom coding to take into account all the operating variables. In *Flexsim*, the production scheduling objects (the line controller and system controller) eliminate the need for that custom code, simplifying the process dramatically.

For simulation purposes, the production environment is defined as follows:

Line: A group of equipment that produces a single output (product) at a time

System: One or more lines that are scheduled as a group

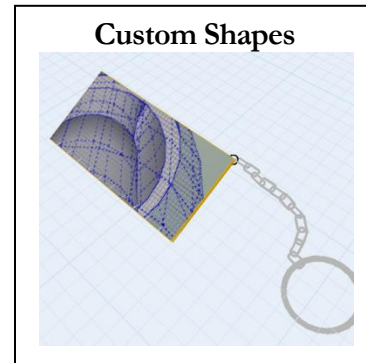


Exercise 15-1 Custom Shapes, Inc

Background

Business customers have long gone to Custom Shapes when they need custom plastic novelty pieces with their own logo or other art work. The Custom Shapes sales team stresses how fast they can produce orders and guarantees that an order will be produced the next business day in the exact order that they are received.

To meet market demands, the plant has worked to create a true pull production system for the custom orders. Orders received during one day are organized by time received and a schedule setup for the next day's production. The plant currently has three stamping lines.



The marketing department wants to expand from the smaller custom orders to larger, longer run orders for the military. Corporate management is thinking of building a new line at another facility. To try to keep the production at the plant, the engineering department has come up with a plan to increase their custom production capacity and take on the new orders by modifying their existing three lines.

Problem statement

Will the engineering department's plan to increase capacity work?



Expected results

- Develop an OFD for the system.
- Simulate the proposed schedule on the current line configuration.
- Simulate the full schedule with the new line configuration.
- Compare the two runs at the end of a 24-hour period.
- Discuss the efficiency of the proposed operation, indicating how the changes could be beneficial.

Modeling and analysis issues

- What time unit is best for this simulation?
- How can you check that each line is running correctly?
- Is there a way to have only one simulation but be able to change the configuration for each run?
- How can you check that the entire simulation was completed?



From Appendix for Chapter 5

Section 1 – Simulation Project Template

<<Simulation project name>>

Date modified:

Current simulation file name:

Simulation software used:

Key Words:

Part I

Problem Definition

Background: *(include project/problem history, reason for simulation, justification)*

Objectives

Key Performance Measures

Key Decision Variables

Simulation Scope

Boundaries

Boundary Assumptions

Operating Assumptions

Part II

Operational Description

System Description

Supporting diagrams, photos, historical data: *(include here, reference, or attach)*

Special logic or other considerations to be included

Conceptual model /Object Flow Diagram (OFD)

⋮

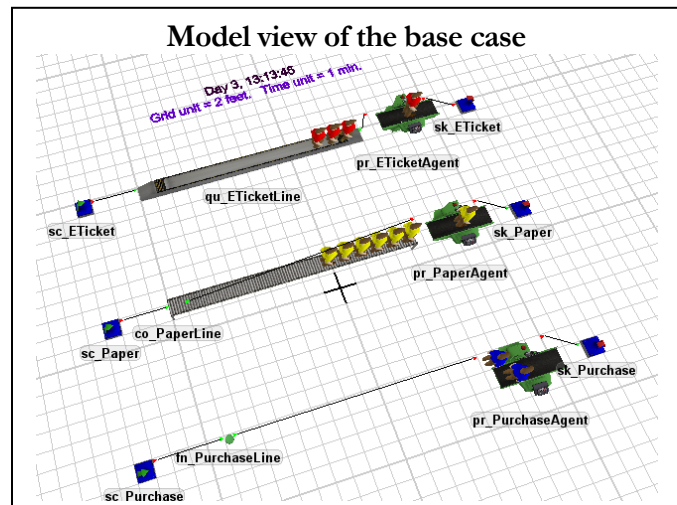
From Appendix for Chapter 6 -- Exercise 6-2 Lucky Air

Notes:

- Select the Person flowitem class from the drop-down menu on the source.
- In order to gain experience with multiple constructs and to compare their capabilities and behavior, use a different mode or object to bring each type of passenger from the source to its agent.
 - Use a queue
 - Change its shape using the General tab to be a long and narrow waiting line
 - On the main queue window select Stack Inside Queue from the dropdown menu for Item Placement.
 - Use a conveyor
 - Use a flow node
- If you assume each grid unit is 2 feet, and time is in minutes, then set the speed on the conveyor and flow node to 125. This means the passengers will move 125 grid units per time unit or 250 feet per minute (standard walking speed of about 3 miles per hour)
- Use the Statistics tab on the various objects to obtain the simulation model results.

It looks strange to see people moving on a conveyor, although a slight change in the graphics and they will look like they are on a moving sidewalk. Similarly, it looks odd that the passengers flow down a machine for the time they spend being served by an agent. Later we'll see how these can be fixed by some tweaking of the graphics; however, for now the focus is on mastering the basic modeling concepts and understanding how the different constructs work. Notice that the objects have been named using the recommended convention; that is,—using the object type abbreviation and a descriptive name.

A completed simulation might look like the following:



From Appendix for Chapter 9 - Modeling Randomness

This Appendix provides instructions on using *ExpertFit* and information on some of the discrete and continuous statistical distributions available in *Flexsim*. It also provides tables for the standard normal distribution and Student's t distribution.

Section 1 Notes for Using *ExpertFit*

ExpertFit not only provides the ability to analyze raw data, but it also allows the user to view statistical distributions and estimate distributions when only general observations are available. The application provides two modes of operation when determining what distribution best fits a data set: Standard Mode, which fits most cases, and an Advanced Mode, which contains additional features for an advanced user. There are also two levels of precision for the data analysis. The *ExpertFit* application contains an extensive Help file as well as tutorials.

The first step in using the application to fit raw data involves collecting a data set to be used for the analysis. *ExpertFit* recommends the following when collecting data for analysis.

- If, at all possible, *collect at least 100 observations* on the random phenomenon of interest, with 200 observations providing more ability to discriminate between two distributions. In general, the benefit from increasing the sample size from 200 to 300 will be less than that provided by increasing the sample size from 100 to 200.
- If collected observations are from a continuous random variable (e.g., a service time), then the data values should have enough resolution so that the sample will have a large number of distinct values. Otherwise, it will be difficult, in general, to find a continuous distribution that provides a good representation.
- If the available data values are integers, then you may want to convert them to real numbers. *ExpertFit* contains many more continuous distributions than discrete distributions.
- You should understand the process that produced the data, rather than treating the observations as just abstract numbers. For example, suppose your data set contains a few extremely large observations;

⋮

About the Authors



Malcolm Beaverstock, Ph.D. After a 40 year career applying advanced control and simulation to industrial problems, Malcolm Beaverstock retired to join Flexsim. During his 13 years with General Mills, he developed and led their simulation program which was involved with more than 300 projects and resulted in significant savings attributed to the use of simulation.

Following graduate studies, Malcolm joined UniRoyal Chemical as manager of their advanced control group where he installed direct digital control systems. During his 17 years at the Foxboro Company he managed various technology groups and established the systems research department that resulted in the early development of Foxboro's I/A control system. As Vice President of Automation Technology, Malcolm initiated work into model-based control applications.

Malcolm holds a Bachelor's degree in Chemical Engineering and Labor Relations from MIT and a Doctorate from Cornell University in Chemical Engineering and Computer Science. He's the author of more than 200 papers on the application and use of advanced technologies.



Allen G. Greenwood, Ph.D., P.E. As Professor of Industrial and Systems Engineering at Mississippi State University (MSU) Allen teaches systems simulation, enterprise systems engineering, and project management. His research interests/expertise include the design and analysis of production and project systems; simulation modeling, analysis, and optimization; and the design and application of decision-support systems.

His professional experience spans a wide variety of domains -- engineering design and development (military aircraft and aerospace), manufacturing and production systems (military aircraft, shipbuilding, automotive, textile fibers, healthcare, and electrical systems), and project management. His work has been funded by such organizations as the Air Force Research Laboratory, Office of Naval Research, Naval Sea Systems Command, NASA, Northrop Grumman Ship Systems, Nissan North America, General Electric Aviation, and Center for Advanced Vehicular Systems (MSU). He has authored or co-authored over 100 creative works, including journal and conference papers, technical reports, software programs, etc.

Allen received his B.S.I.E, M.S.I.E, and Ph.D. (Management Science) degrees from North Carolina State University, University of Tennessee, and Virginia Tech, respectively. Prior to joining MSU, he held positions at American Enka Company, General Dynamics Corporation, Virginia Tech, Northeastern University, and the American University of Armenia.



Eamonn Lavery, Ph.D. Eamonn completed his first object-oriented simulation software while in graduate school. In 1999 he was hired by F&H Simulations B.V. to conduct research and development for new simulation products. He developed the OpenGL virtual reality animation views for Taylor ED simulation software. In 2003 Eamonn joined Flexsim Software Products, Inc. as the Chief Technology Officer. He is the author and architect of the Flexsim Software.

Eamonn holds a Bachelor of Science in Mechanical Engineering and a Doctorate in Object Oriented Modeling and Simulation of Manufacturing Systems from Queens University of Belfast.



William Nordgren, MS CIM. Bill is the President and CEO of Flexsim Software Products, Inc. In 1988 he founded ProModel Corporation and was Vice President until he left in 1992. In 1993 Bill founded F&H Simulations, Inc. (now Flexsim Software Products, Inc.) and introduced Taylor II, Taylor ED, and Flexsim into the market.

Bill has authored several papers dealing with simulation project management, queuing theory, and has taught hundreds of classes in the use of simulation software. He is listed in Marquis Who's Who in America for his accomplishments in the advancement of simulation technology. Bill received a Bachelor of Science in Manufacturing Engineering Technology, and a Master of Science in CIM (Computer Integrated Manufacturing) from Brigham Young University.
