

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 Models105

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 Data131

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 Tables143

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 Framer.	158

Chapter 9 Modeling Randomness163

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 <i>FlexSim's</i> Experimenter	196
Section 10-5	Comparing two alternatives	198
Section 10-6	Variance reduction through common random numbers	200
Section 10-7	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	272
Exercise 14-1 James Peanuts	275
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	326
Section 4 Exercise 7-2 Even More Lucky Air	332
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 Framer	343

Appendix for Chapter 9345
Section 1 Notes for Using <i>ExpertFit</i>	345
Section 2 Statistical distributions in <i>FlexSim</i>	346
Section 3 Tables of normal and t distributions	351
Appendix for Chapter 10355
Section 1 Setting up the Experimenter	355
Section 2 Defining scenarios	357
Section 3 Advanced experimenter options	358
Appendix for Chapter 11359
Section 1 Exercise 11-1 Kegglers Brew	359
Section 2 Exercise 11-2 Chairs for Tots	360
Appendix for Chapter 12363
Section 1 Commonly-used commands	363
Section 2 FlexSim software architecture	367
Section 3 Exercise 12-1 Hilltop Steel	371
Appendix for Chapter 13375
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 14383
Section 1 Exercise 14-1 James Peanuts	383
Section 2 Exercise 14-2 Western Grain	385
Appendix for Chapter 15387
Section 1 Custom Shapes Exercise	387
Appendix - Interacting with Other Applications391
Section 1 Microsoft Excel import/export	391
Section 2 Importing AutoCAD drawings	392
Appendix - Advanced Techniques395
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 Applications407

Section 1 Health Care	407
Section 2 Container Terminal (CT).....	408
Section 3 Express It 3D presentation software	409
About the Authors	411

Chapter

6

Building Basic Simulation Models

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.

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.

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

Section 6-1 Simulation environment

FlexSim, like other professional simulation software, provides extensive support for building and analyzing simulation models. Figure 6.1 outlines the basic components found in such software. Most of the components, especially those in the left portion of the figure, will be discussed later in the book.

This chapter concerns the modeling environment and focuses on the components in the right portion of Figure 6.1. All simulation software provides a set of pre-defined modeling objects that facilitates model building. Such objects include queues that store items awaiting processing (due to unplanned delays), processing objects that modify or create planned delays for items as they flow through the model (service operations), transporting objects that move items through the model, etc. The number and capability of such objects varies widely in the various simulation software products that are on the market today. Some software, like *FlexSim*, allows users to easily change the behavior of objects, and Advanced Users can create their own objects.

The simulation environment for most software applications is similar and follow generally accepted conventions.

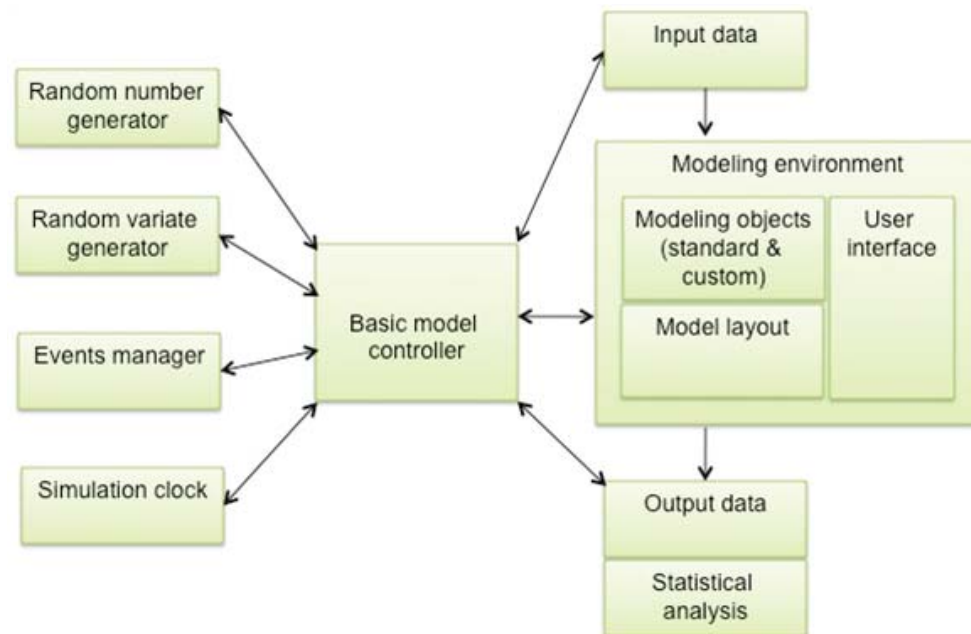


Figure 6.1 Basic components of simulation software

The modeling objects are typically placed and arranged on a layout, or simulation surface, that resembles the physical layout of the system being modeled. The objects are connected to represent the flow of items through the model and permit communication among objects.

All simulation software products have their own user interface that enables users to effectively and efficiently build models, import data, analyze results, etc. While much of the underlying logic is the same across the various simulation software products, the user interface and accessibility to the underlying logic varies exten-

sively. *FlexSim*'s interface is quite effective for all levels of users; however, much of its power lies in its openness and ability to modify the objects and their behavior to best match the modeling requirements.

As mentioned above and shown in Figure 6.1, a simulation model is data driven; that is, without data, the model is useless and cannot function. Simulation software, therefore, needs to provide mechanisms to easily import data. Similarly, the reason we build and execute models is to obtain information and to understand the consequences of actions; therefore, simulation software needs to provide mechanisms that easily present or export data and that facilitate the analysis of that data.

Simulation environment terminology

When describing a simulation, certain terms are used to talk about the component pieces. While the specific names may vary in each software application, the concepts and functionality are basically the same.

Time and space

Time and space in the simulation environment are dimensionless until the user decides what the dimensions should be. For some reports, *FlexSim* provides a default set of units: Seconds, Meters, and Liters. To change the defaults go to the File/Global Preferences/Environment and select the desired units. *FlexSim* then has to be closed and re-opened for the change to take place. In the same place you can choose to have the screen displayed whenever a new model is created.

When starting a new model, a screen appears as shown in Figure 6.2, showing the definition of units. The values are used during some reporting and internal calculations. All data input for the model should be consistent with these units. For example, if the time unit is seconds, the time between arrivals and cycle time for a processor would both have to be expressed in seconds. As result, statistics that are generated by the simulation, such as the amount of time a processor is operating, will be interpreted in seconds. Once selected, the units may not be changed until a new model is opened. The units for a model are saved with the model file.

Space is also dimensionless. The default unit of space on the simulation layout screen one grid unit of size 1x1x1. The user has to decide the actual units that the default grid space

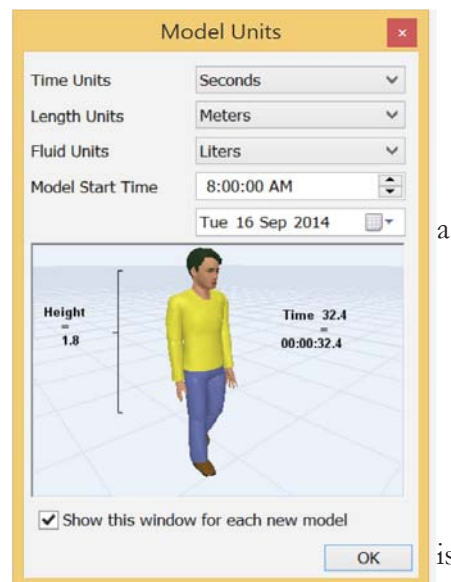


Figure 6.2 Units screen

represents. Equipment can be scaled appropriately. Task executors, such as fork trucks, have speed values assigned to them as the number of grid units traveled per clock unit. Being consistent with data values is critical to a successful simulation.

The main screen

Chapter 3 dealt with models that were already built along with the simulation environment to run them. This section starts with a blank page or layout known as the main screen as shown in Figure 6.3.

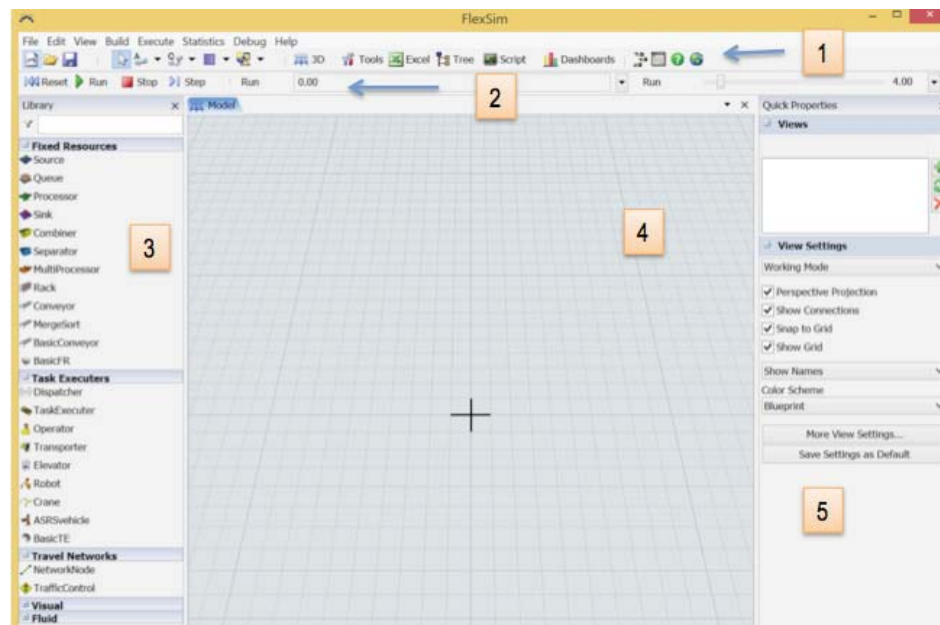


Figure 6.3 *FlexSim* model development main screen interface

As discussed in the previous section, all simulation applications have an environment in which a simulation model is built. That environment contains the surface for building the model (sometimes referred to as a layout), menus for controlling files and functions, and basic units or objects that are used to build the simulation model. *FlexSim*, as with most other simulation software, uses the conventions associated with Microsoft applications (file management, drop-down menus, tool bars, etc). As a side note, in the student version of *FlexSim*, the watermark “Educational Version” appears on the layout.

The main screen is made up of five major areas:

Area 1: Main toolbars

- Top row: Set of Microsoft Windows style drop-down menu options for file control as well as *FlexSim* operating controls
- Lower row: Shortcut buttons that allow quick access to some common *FlexSim* interfaces, operating elements, and model views

Area 2: Simulation control panel

- Buttons
 - **Reset:** Initializes the model
 - **Run:** Begins execution of the simulation
 - **Stop:** Stops the model at the end of the current clock cycle (A simulation can be re-started from the stop point.)
 - **Step:** Moves simulation ahead to the next scheduled model event
- Time controls
 - **Run Time:** Displays the current model time in simulation time units
 - **Stop Time drop down arrow:** Sets a specific time at which the simulation will be stopped (Once stopped, the value can be changed to set a new future stop time)
 - **Speed Slider:** Defines the number of simulation time units per second of real time. The drop down arrow can set a specific rate.

Area 3: Library icon grid

- Objects that can be brought into the simulation model by clicking and holding down the mouse button while dragging the object onto the modeling surface or layout. Other libraries can be added.

Area 4: Model view window

- Surface where the simulation model is built

Area 5: Quick Properties pan

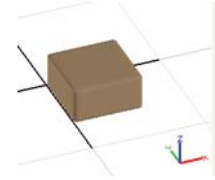
- Displays that can be used to monitor and change data for various simulation objects. The displays will change based on the particular context.

Most applications allow modifications to the basic screen. The Appendix shows how the *FlexSim* simulation workspace can be visually changed.

Section 6-2 Simulation components

Graphically-oriented simulation software applications make use of elements that are placed on the simulation surface and can be viewed. In *FlexSim*, an **object** is the most basic building element of a simulation.

In *FlexSim* there are two basic types of objects: discrete and fluid (continuous). The discrete objects are used to develop discrete-event simulation models where model behavior results from events that occur at discrete points in time, such as an item arriving to the system or a machine stopping due to an internal failure. Fluid or continuous objects are used to model behavior that results from changes that occur continuously over time, such as the filling of a tank with a liquid. Most of the focus of this book is on discrete-event models and the use of discrete objects. Fluid objects are discussed in Chapter 14.



Simulation objects perform specific functions while flowitems move within the simulation.

Simulations normally involve actual discrete entities that physically move around in the simulated environment. In *FlexSim*, these entities are called flowitems. Depending on the simulation, these could be boxes, products, customers, paperwork, and so on. Without flowitems, there isn't a need for simulation. Some simulation software packages refer to flowitems as entities or transactions.

In *FlexSim*, flowitems are listed in the Flowitem Bin which is accessed through the Tools menu at the top of the screen. Normally, flowitems are brought into a simulation through the source object where the choice of using a 3D flowitem is listed in a drop-down menu. Additional details are included in the Appendix.

Simulations also need objects that interact with flowitems. These might perform an operation, create a delay, or move the items. In *FlexSim*, there are two general categories of such discrete objects: fixed resources and task executors (mobile resources).

A simulation model is simply a collection of these objects put together in such a manner as to simulate the behavior of a system. The *FlexSim* simulation model shown in Figure 6.4 is one such combination of objects.

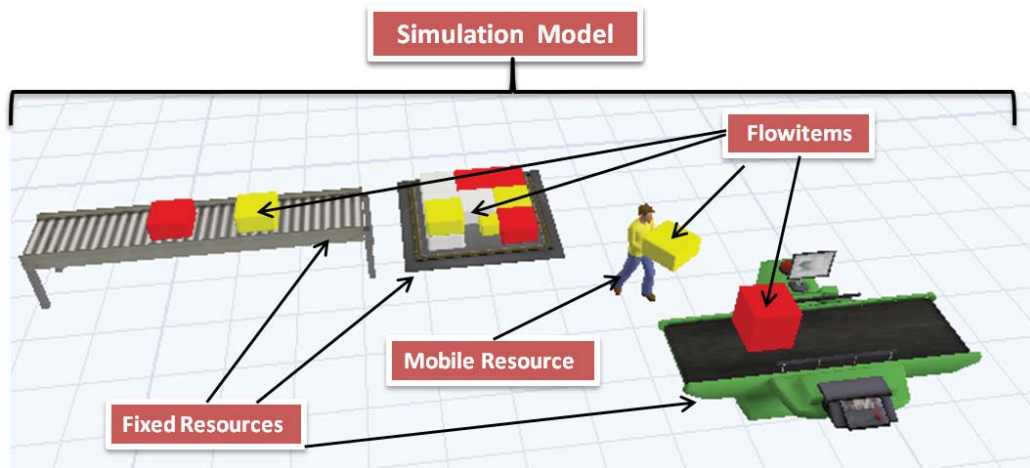


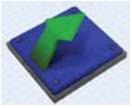
Figure 6.4 *FlexSim* simulation terminology

Section 6-3 Fixed resources

Fixed resources are the objects that send, receive, and perform activities/operations on flowitems. They are also the most common object type. They are referred to as “fixed” because they are largely stationary. Once they are placed on the model surface, they tend to stay in that place unless manipulated later by the modeler.

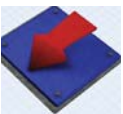
Examples of this type of object include the following:

Source



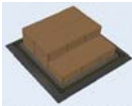
- Creates and releases flowitems
- Modes of arrival: interarrival time, arrival schedule, arrival sequence

Sink



- Receives and removes flowitems from the simulation

Queue



- Temporarily stores flowitems when downstream objects cannot accept them
- Can receive multiple flowitems at a time
- Can batch process flowitems
- Receives flowitems until its specified maximum content is reached

Processor



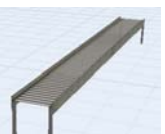
- Processes or forces a delay on a flowitem
- May call operators for setup or process operations
- May incur scheduled or unscheduled downtimes
- Handles one flowitem at a time or multiple flowitems independently (if its maximum content > 1)

MultiProcessor



- Performs a set of operations or processes in sequence
- Operations/processes may have separate times and call separate resources
- Handles one flowitem at a time

Conveyor



- Moves flowitems over a fixed path (not necessarily linear) at a specified speed
- Flowitems enter and leave the conveyor one at a time
- Modes: accumulating, non-accumulating

Objects that normally stay in one position and perform operations on a flowitem are called fixed resources.

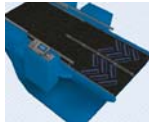
- Capacity limited by number of flowitems or available space on the conveyor. Spacing between flowitems may be specified.
- The library may contain many types of conveyors that allow advanced functions such as defining when and how flow items leave the object.

Combiner



- Groups multiple flowitems
- One input from Port 1 which may be a container to hold flowitems from other ports
- Can have process times, use resources, and incur scheduled or unscheduled downtimes
- Modes: join, cannot be separated; pack, container from port 1 holds others—can be separated; and batch, all objects travel as a group.

Separator



- Accepts one input at a time.
- Can have process times, use resources, and incur scheduled or unscheduled downtimes.
- Modes: unpack from packed container or split, makes copies of a flowitem.

Section 6-4 Transporting items



The transportation of flowitems in a *FlexSim* model is the physical movement from one resource to another. That movement may occur in a number of different ways. By default, fixed resource objects in a model pass flowitems to each other instantly. In addition, flowitems may move from one fixed resource to another through the use of an intermediary fixed resource object, such as a conveyor.

Flowitems may also be moved by **task executors**. These are special objects that have mobility within a model. As the name suggests, they are free to move about within the model executing tasks that have been assigned to them. They may transport flowitems or be used as a resource for fixed resource. Examples of task executors which move flowitems are a fork truck, crane, ambulance, wheelchair, or person. Task executors that move to fixed resources and perform other tasks include an operator required for some setup, processing, or maintenance operation.

All of the types of fixed resources, task executors, and other objects are shown in the library grid in Figure 6.5. The grid is located on the left side of the main *FlexSim* interface.

Task executors move flowitems from one place to another.

Data, used in simulation logic, can be placed on all *FlexSim* classes.

Exercises

Simulation is an applied technology and has little meaning when used to simply create models without an objective in mind. The belief that if you just build a simulation something good will happen doesn't hold true in practice. Chapter 4 of this book emphasized the need to establish a proper scope and detail level for simulation projects and discussed a methodology.

Lessons learned by actually solving problems using simulation are more likely to be remembered. The exercises in this and later chapters follow this practice. In each case the problem background and associated data are provided. The steps to follow should include an analysis of the system, a definition of how best to simulate the system using an OFD for planning, and an understanding of what simulation functions are required to resolve the problem.

Exercise 6-1 Johnson Pharmaceutical

Background

As the largest supplier of over-the-counter medications, Johnson is redesigning its cold remedy line for higher capacity in time for the upcoming flu season. The plan is for the packing line to be completely re-designed. The new plan, conceived by the corporate packaging group, is based on the “greedy” concept. In that design, boxes of individual doses are taken off the main conveyor line by the first test/wrap machine.

A second test/wrap machine takes off boxes further down the conveyor. The boxes arrive in the packing area at an average rate of 60/min. The wrap machines are rated at an average of 40/min. Based on average rates, the first two machines can handle 133% of the incoming boxes rate; however, a third will also be added as a backup. The plant engineering manager has asked you to validate the design before the plant commits to the proposed production rate.



Figure 6.19 Pharmaceutical packaging

Problem statement

Validate the new packaging line design.

Operating data

Time between arrivals of medical boxes: exponentially distributed (0,1,1) with mean of 1 second. The first parameter is the location parameter and is usually 0; the second parameter is the mean; the third parameter identifies the random number stream to use (more on setting this value later).

- Test/Wrap machine cycle times: exponentially distributed (0,1.5,2) with a mean of 1.5 sec.
- Conveyor speed: 1 unit/sec.

Expected results

- Draw an OFD for the system
- Comment on the design by running a simulation for an 8-hour shift (28,800 sec). What would you recommend to the plant manager? What rate should the plant commit to?

Modeling and analysis issues

- How should the conveyors be shown? What is the flow logic for them?
- Based just on the average rates what would you expect the capacity of the line to be and why?

Simulation results and comments are included in the Appendix.

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



Figure 6.20 Lucky Airlines

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.

Operating data

Estimates of passenger demand, in terms of times between arrivals to the ticketing area and time for an agent to service each passenger type are provided in the table below. Assume the time between arrivals is exponentially distributed while agent service time is normally distributed—all times are in minutes.

Passenger type	Time between arrivals	Service time
E-ticket	mean: 5 min	mean: 3 min std. dev. 1 min
Paper ticket	mean: 10 min	mean: 8 min std. dev.: 3 min
Purchase	mean: 15 min	mean: 12 min std. dev.: 3 min

Expected results

- Fill out Parts I and II of the project template.
- Provide the following metrics based on a simulation run of 168 hours:
 - The average and maximum length of the lines for each agent's station
 - The average wait time for each type of customer
 - The average number of customers each station services per hour (throughput)

- The average utilization for each agent
- Prepare an executive summary based on conclusions drawn from the simulation.

Modeling and analysis issues

- 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 passengers to the agents? Does the choice significantly impact the results?

Simulation results and comments are included in the Appendix.

Review questions

1. Identify three “nuggets”—the things you found to be the most interesting or most important—in the chapter.
2. Discuss the components included in the modeling environment of most simulation software applications.
3. Identify and define the basic elements that comprise a simulation model developed in *FlexSim*.
4. Describe how the movements of flowitems between fixed resources occur within a *FlexSim* model.
5. Describe the difference between fixed resources and task executors (the two basic categories of resources used in *FlexSim*).
6. Discuss how the *FlexSim* object structure enhances model development.
7. Select several *FlexSim* objects; for each one, identify the various states that the object can be in during a simulation.
8. Compare the behavior of accumulating and non-accumulating conveyors.

9. Construct a *FlexSim* model of the Compu-Help system described in the Chapter 5; however, assume the devices do not crash. Run the model for 10 hours and determine the following:
 - a. Average number of calls waiting
 - b. Average time a caller waits before being helped
 - c. Percentage of customers lost due to the system's limitation of only being able to keep six calls waiting.
10. Modify the basic Compu-Help model so that the maximum queue capacity is 1000. Run the model for 5 hours, stop it (do not reset), and note the average wait time. Run the model for 5 more hours, stop it (again, do not reset), and note the average wait time. Repeat this eight more times. Plot the average wait time versus cumulative run time. What can you infer from the graph?
11. Discuss how a model's behavior differs when a queue, conveyor, and flow node are used to move flowitems between two fixed resources.

Index

A

Advanced User 14, 16, 17, 18, 166, 225, 227, 260, 313
AGV 77, 149
Analyst 68, 105, 165, 244
Arrival sequence 25, 48, 55, 110, 335
Arrival style 335
ASRS 150, 395
Assumption 23, 39, 67, 76, 79, 95, 198, 206
Attribute 89, 112, 114, 115, 120, 131, 132
Availability 30, 34, 72, 207, 209

B

Batch 25, 27, 28, 29, 34, 58, 111, 145, 272, 274, 275, 276, 277
Bottleneck 12, 13, 26, 32, 56, 158, 209, 248, 276
Boundaries 79
Boundary assumptions 79, 95
Breakdowns 5, 25, 46, 85, 89, 100, 206, 215, 222, 360, 361, 372
Breakdown tab 114, 212, 213, 214
By expression 122, 123, 134, 137, 328, 331

C

C++ 233, 234, 367, 370
CAD 81, 96
Centerobject 237, 363, 373
Center port 16, 116, 119, 154, 215, 231, 336, 337, 363
Changeover 81, 287, 387, 388
ChangeOver Table 287
Cleanout Time 287
Code edit 313, 392
Code template 239, 312
Combiner 111, 143, 144, 145, 146, 149, 160, 161, 214, 215, 256, 257, 271, 335, 360, 361, 378
Competing failure 210, 211, 215
Complexity 4, 25, 27, 29, 32, 72, 76
Conceptual model 67, 68, 73, 75, 318
Connections 81, 114, 116, 117, 118, 119, 120, 122, 136, 154, 214, 237, 285, 287, 310
Container 8, 83, 97, 111, 144, 145, 146, 155, 178,

217, 236, 237, 272, 366, 408, 409

Content 85, 120, 122, 123, 154, 206, 238, 240, 244, 248, 272, 343, 361, 365, 377, 380, 409

Controller 17, 253, 254, 283, 284, 285, 286, 287, 288, 289, 293

Conveyor 80, 87, 111, 112, 114, 119, 120, 121, 122, 153, 229, 255, 284, 309, 313, 314, 315, 316, 336
Conveyor layout 122, 313
Conveyor object 119, 315

Copy From Highlighted 310

Creation time 378, 381

Current object 134, 135, 137, 237, 240, 244, 245, 248

Custom code 206, 230, 233, 239, 241, 283, 324, 372, 392

Custom logic 16, 17, 121, 136, 227, 233, 236, 238, 244, 260, 275, 286

Cycle time 35, 109, 126, 165, 228, 248, 254, 288, 371, 372, 377, 379, 387

D

Data engine 368, 369

Data nodes 229, 230

Decision support 5, 6, 19, 48, 65, 68, 70, 71, 73

Decision Support 71

Decision variables 47, 49, 124, 187, 189, 191, 202, 317

Degree of complexity 25, 32

Dempirical 165, 332, 347, 380

Designer 68

Design of experiments 91, 367

Developer 68, 90, 92, 131, 169

Discrete empirical 165, 178, 333, 347

Discrete event 23, 33, 34, 35, 269, 273, 281

Discrete object 109, 110, 149, 272, 273, 281

Dispatcher 153, 215, 337, 396

Domain knowledge 69

Down function 215, 216, 361

Downtimes 30, 31, 55, 80, 85, 91, 137, 156, 208, 209, 210, 211, 212, 213, 214, 215, 216, 341, 388

Drop-down menu 105, 108, 109, 113, 118, 122, 134, 135, 213, 215, 242, 271, 298, 310, 316, 324, 358, 391, 392, 398, 399, 403

E

Editing objects 114
Educator 69
Efficiency 3, 6, 12, 28, 56, 60, 94, 95, 205, 292, 299, 361, 374, 407, 408
Errors 12, 202, 233, 239, 240, 241, 242, 243, 290, 330, 346
Experimenter 47, 187, 196, 197, 203, 355, 357, 358, 360, 361, 373
ExpertFit 68, 165, 166, 167, 168, 175, 345, 346

F

Failure clock 207, 210, 211, 212, 214, 215, 217
Failure mode 32, 210, 211, 215, 217
FIFO 158, 176, 177, 180, 182, 183, 189, 199, 200, 201, 237
File menu 297
First available 119, 135, 136, 271
Fixed resources 84, 85, 86, 100, 110, 112, 118, 120, 122, 129, 143, 150, 319, 340, 395
Flag 25, 372, 377, 380, 381, 384, 385, 388
Flexscript 230, 233, 234, 235, 238, 242, 244, 245, 247, 248, 324, 363, 370
Flexscript commands 230, 235, 238, 244, 245, 247, 248, 363
Flowitem 109, 110, 119, 121, 122, 132, 143, 144, 145, 146, 254, 272, 323
Flow logic 126, 136, 146, 320, 331
Flow node 120, 122, 128, 129, 153, 316, 379, 380, 381
Fluid 269–281
Fluid blender 271, 281
Fluid flow 35, 269, 270
Fluid Generator 271
Fluid mixer 272, 274, 281
Fluid objects 109, 270, 271, 272, 273, 274, 280, 281, 320, 383
Fluid pipe 271
Fluid Processor 272
Fluid splitter 272
Fluid tank 271, 272, 386
Fluid terminator 271
Fluid to item 272
Functional specification 45, 77, 79, 81, 90, 91, 92, 93

G

General tab 115, 116, 120, 155, 258, 259, 311, 312, 315, 316
Global table 139, 165, 213, 245, 246, 247, 248, 256, 257, 320, 332, 333, 347, 348, 359, 360, 383
Global variable 242, 243, 245, 246
Graphical editor 156, 341, 342
Grid 108, 109, 112, 120, 151, 152, 153, 271, 300, 310, 316, 319, 338, 400
GUI 70, 71, 228, 298, 300, 398, 399, 400, 401, 402

H

Health care 9, 31, 407, 408
Highlighting 139, 233, 310
HighMark 383, 384, 385

I

Ideal rate 205
Implementer 69
Input port 118, 119, 144, 161, 237, 248, 254, 285, 287, 342, 363, 377, 378
Interface control 297, 300
Intermediate User 15, 16, 18, 103, 105, 143, 227, 233, 256, 313
Investigator 69
Involved object 339, 397
Item to fluid 272
Itemtype 132, 133, 135, 136, 142, 183, 238, 325, 326, 329, 330, 331, 332, 343, 360, 364, 377, 380, 404

J

Job shop 155, 203, 340
Join 30, 55, 88, 111, 145, 317, 318

L

Labels 18, 112, 114, 131, 132, 133, 230, 238, 256, 310, 319, 323, 328, 365, 371, 376, 377, 378, 380, 381, 400
Label-table 256, 257, 261, 263, 376, 378
Language engine 368, 370
Lean 2, 3, 11, 17, 18, 25, 26, 30, 32, 38, 39, 41, 71, 216, 217, 275, 277, 407
Level indicator 273
Library 17, 18, 108, 112, 113, 149, 155, 251, 260, 261, 271, 369, 370, 385, 407, 409

Library icon grid 108, 271
Line controller 283, 284, 285, 286, 287, 288, 289, 293, 387, 388
Logic 23, 67, 80, 82, 89, 114, 131–142, 146, 153, 227, 230, 238, 240, 244, 255, 275, 331, 388

M

Main screen 107, 108
Main toolbars 108, 113
Manufacturing cell 203, 252, 261
Material distribution 30
Maximum rate 205
Members 213, 214, 215, 287
Message 83, 85, 132, 133, 137, 240, 243, 251, 252, 253, 254, 366, 377, 386
Message trigger 132, 366
Methodology 60, 61, 64, 75, 76, 82, 87, 118, 125, 163, 169, 188, 202, 322
Metrics 23, 25, 29, 37, 45, 46, 49, 51, 52, 79, 95, 127, 138, 141, 205, 206, 209, 263, 280, 383
Midmark 383, 384, 385
Mission length 207
Mixer recipe 274
Mixer steps 274
Mobile resources 84, 85, 100, 110, 122, 135, 143, 149, 161, 395
Modeling 1, 21, 22, 23, 25, 47, 63, 65, 66, 67, 68, 72, 81, 86, 105, 106, 110, 198
Modeling paradox 40
Modeling utilities 261, 309, 310
MTBF 85, 207, 208, 209, 211, 212, 213, 214, 215, 217, 219, 248
MTTR XI, 85, 207, 208, 209, 211, 212, 213, 214, 215, 217, 219, 248
MultiProcessor 111

N

Naming convention 114, 140
Network layer 367, 370
Network node 149, 153, 154, 158, 337, 338, 339, 343, 360, 380, 386
No Connect 338, 339
Node 120, 122, 128, 129, 149, 153, 154, 158, 228, 229, 230, 231, 234, 235, 236, 237, 238, 243, 247, 257, 288, 316, 337, 338, 339, 343, 360, 363, 365, 367, 375, 376, 379, 380, 381, 386, 396, 400, 401, 402, 403

Data node 229, 230
Flow node 120, 122, 128, 129, 153, 316, 379, 380, 381
Tree node 230, 243
Variables node 230, 238
No Passing 338, 339

O

Object attributes 364
Object control 366
Objective 25, 61
Object labels 365
Object nodes 229, 230, 236, 237, 238, 247
Object oriented 34
Object spatial attributes 364
Object statistics 15, 123, 124
Object variables 274, 367
Occasional User 13, 14, 15, 16, 19, 43, 45, 46, 48, 60, 61, 76, 92, 297
OFD 14, 15, 82, 83, 86, 87, 100, 125, 126, 138, 148, 158, 160, 263, 277, 280, 292, 307, 318
OnCreation trigger 232, 325, 326, 332
OnEntry trigger 133, 137, 182, 231, 236, 245, 247, 257, 259, 288, 324
OnExit trigger 133, 137, 182, 232, 238, 288, 320, 325, 326, 329, 359, 376
OnReset trigger 133, 243, 246, 253, 288, 371
OpenGL 367, 370, 410, 412
Operability 39, 90
Operating assumptions 79
Operations system 4, 5, 19, 24, 26, 38, 40, 41, 61, 63, 120, 143, 160, 161, 185
Output port 118, 119, 149, 232, 237, 272, 273, 327, 331, 363, 397

P

Pack 145
Passing 153, 338, 339
Performance 5, 12, 13, 408
Performance measure 14, 16, 17, 23, 37, 47, 78, 85, 87, 91, 92, 101, 123, 165, 177, 181, 188, 189, 190, 191, 192, 193, 194, 195, 197, 199, 202, 205, 206, 266, 278, 305, 317, 318, 322, 358, 361, 409
Picklist 115, 134, 136, 144, 146, 154, 233, 238, 275, 311, 312, 313, 324, 326, 332, 333, 335, 375
Picklist options 115, 134, 144, 146, 233, 238, 312, 313, 326, 332, 333
P&ID 81

- Planning rate** 205
Port rate 272
Preempt 215, 216, 337
Priority 40, 58, 140, 141, 150, 156, 157, 158, 215, 216, 248, 319, 333, 337, 342, 396
Probabilistic 124, 163
Probability distributions 23, 46, 68, 163, 165, 166, 168, 169, 172, 173, 175, 176, 178, 179, 181, 182, 347, 348
Procedure 63, 80, 82, 90, 95, 97, 133, 159, 178, 260, 264, 265, 266
Processor 111, 134, 231
Product Data 286, 287, 388
Production line 26, 30, 37, 79, 92, 269, 283, 284, 286, 287
Production schedule 14, 25, 29, 60, 95, 283, 284, 285, 293
Programmed model 67, 73
Project template 77, 90, 92, 93, 94, 98, 127, 316
Properties window 115, 131, 394
Pseudocode 245, 247
Pull 3, 31, 32, 84, 135, 137, 142, 289, 381
Pull production 289
- Q**
Queue object 110, 119, 325, 401
Queuing theory 23, 24, 87, 192, 412
- R**
Recorder 251, 257, 375, 376, 378
Referencing objects 236, 363
Reliability 17, 27, 32, 60, 72, 91, 114, 206, 207, 209, 210, 213, 214, 215, 283, 291, 409
Repair time 96, 165, 185, 207, 208, 209, 210, 211
Researcher 69
Reset 49, 113, 129, 133, 194, 245, 246, 297, 298, 299, 312, 339, 372, 377, 380, 383, 385, 388, 403
Resources 319, 320
Round robin 136, 142
Run Time 58, 79, 108, 129, 297
- S**
Screen elements 154
Selecting 49, 51, 70, 116, 136, 165, 239, 243, 256, 260, 299, 310, 312, 323, 336, 337, 357, 397, 400, 403
- Senddelayedmessage** 251, 252, 366
Sendmessage 251, 252, 366
Send to Port 135, 136, 142, 232
Separator 111, 146, 161, 272
Service operations 1, 8, 106
Set itemtype 332
Simple cell 86
Simulation benefit categories 10
Simulation control panel 108
Simulation engine 369
Simulation knowledge 69
Simulation languages 33, 34, 251
Simulators 7, 22, 33, 34, 35, 46, 83
Sink 110, 122, 212, 217, 229, 245, 246, 330, 360, 378, 380, 381, 388, 398
SMA life cycle 65
SMA process 66, 75
SMA project 65, 68, 69
Source 71, 83, 110, 219, 232, 271, 404
Space 109
Speed control 298, 373, 374
Speed Limit 338
SQL 367
Staffing 3, 13, 28, 39, 48, 50, 51, 52, 96, 264, 407
Stakeholders 11, 45, 64, 65, 67, 69, 72, 73, 75
State 37, 71, 123, 403, 404, 411
Statistics tab 49, 117, 123, 257, 316
Staytime 123, 206
Stochastic 22, 46, 47, 61, 124, 163
Stop time 51, 108, 244, 298
Subnodes 229, 230, 236, 376
Surge 26, 32, 216
Switch 233, 240, 241, 252, 378
System controller 17, 283, 284, 285, 286, 387, 388
- T**
Target 144, 205, 206, 248, 288, 346
Target rate 205
Task executers 112, 122, 129, 143, 149, 150, 153, 215, 216, 320, 335, 336, 337, 339, 380
Throughput 13, 48, 58, 78, 87, 94, 117, 128, 146, 182, 199, 200, 201, 203, 217, 322, 361, 401, 408, 409
Ticker 270
Time controls 108
TIS 245

Tools tab 155
Tree 15, 18, 228, 229, 230, 234, 235, 236, 237, 243,
257, 288, 357, 375, 376, 387, 398, 400, 401, 402,
409, 410
Tree node 231
Triggers 114, 132–142, 182, 231, 253, 256, 273, 310,
358, 383
 Flow trigger 134, 142, 328, 331, 372, 381
Trigger tab 132, 142, 273
Troubleshooting 243

U

Unpack 146, 161
User 7, 13, 14, 15, 16, 17, 18, 19, 43, 45, 46, 47, 48,
60, 61, 76, 89, 92, 103, 105, 138, 143, 150, 155,
166, 225, 227, 233, 242, 256, 257, 260, 271, 297,
298, 313, 321, 340, 346, 368, 370, 375, 376, 397,
398, 399, 403
User interface 89, 370
User interface engine 370
User library 251, 260, 261
Use transport 362
Utilization 11, 12, 23, 56, 78, 85, 87, 93, 128, 138,
141, 203, 322, 331, 333, 407, 408, 409

V

Validation 32, 39, 40, 73, 90, 91
Value stream map 14, 81, 82
View Settings 309
Virtual distance 153, 343, 360, 380
Visual Display 154, 311, 394
Visualization 120, 143, 151, 152, 154, 227, 228, 258,
278, 367, 370, 381, 409
Visual tool 154, 155, 245, 246, 247, 253, 254, 261,
311, 381, 389
VR 368, 370

W

Wizards 105
Workstation 33, 86, 87