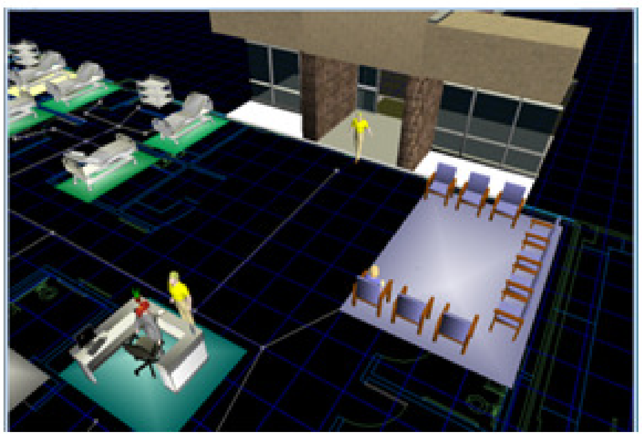
Applying Simulation with FlexSim

3D SIMULATION SOFTWARE

Learning to get the Most Out of Simulation Technology



Preview Edition

Malcolm Beaverstock, PhD
Eamonn Lavery, PhD
William Nordgren MS CIM

FLEXSIM SIMULATION SOFTWARE

Applying Simulation with Flexsim – Preview Edition

This special edition provides excerpts from the full text book along with sample exercises.
The continuance icon

Indicates material that appears in the final edition but not included in this Preview Edition

Canyon Park Technology Center
Building A – Suite 2300
Orem, Utah 84097 USA
Phone (801) 224-6914 • Fax (801) 224-6984
sales@flexsim.com
www.flexsim.com

Preface:

Using simulation is the best way to learn about it. This book is intended to provide an immersion experience into using simulation for the analysis of dynamic systems. The book can be used in a classroom setting or by individuals.

When used as the basis for teaching simulation, this text provides material for meeting course objectives at various levels. Classroom time may focus on a combination of the basic concepts of the first section, review of specific exercises, or topics in the appendix covering statistical distributions, advanced techniques, or simulation software architecture.

The book is based on three principles taken from years of experience in using, and teaching, simulation.

First: *Simulation is an essential tool for use by ALL individuals who are involved with understanding, analyzing, and interpreting dynamic systems.*

Second*: The full value of using simulation is gained by those actively taking part in the specifying, developing, and using simulation.*

Third*: Visualization through simulation provides critical information needed for communicating and understanding the simulation analysis.*

The text is organized as a workbook, allowing individuals to use simulation technology immediately. All exercises, based on examples from manufacturing, service, and other areas, define an operational problem to be solved or an issue to be resolved. Simulation is used to provide insight and as a basis for presenting solutions or options. The progressive series of exercises makes this book a hands-on experience for users with varying levels of expertise or need of simulation.

This book is not a manual for any particular software. Modern software applications should already contain detailed instructions within product tutorials, help files, documentation, and training. Flexsim software is used throughout the book because of its ease of use and exceptional visualization functionality. The Flexsim application and its underlying software structure allow it to be easily adaptable to the diversity of exercises and user involvement levels discussed in the book. A fully functional student edition is available with the text.

The book's First Section reviews the history of simulation. Economics and technology have changed both simulation applications and the needs of individuals wanting to use simulation. This section provides examples across a wide variety of human, machine and other boundaries that benefit most from the use of simulation. It identifies how simulation interacts and complements business, lean, value stream, organizational, and other improvement tools. Economic justification of simulation and examples of simulation's value are also provided. Finally, the various levels of user interaction with simulation are defined and become the basis for the remainder of the book.

The Second Section helps those who will only have an occasional interaction with simulation. These individuals may need to quickly open a prepared, even complex, simulation and carry out specific scenarios. Through a series of exercises, users learn how to open a simulation, validate or change settings within the simulation, and recover the resulting statistics. The occasional user, who needs to specify a problem or operation to be simulated by others, will learn how to correctly establish a simulation project. Exercises show how to define a Dynamic Functional Diagram (DFD) based on simulation elements. Such a diagram then serves as a functional specification, documentation, and starting point for the simulated system.

The Third Section develops skills for Intermediate users. Those users, while not simulation experts, will be involved in building and adding details to simulations. Exercises in this section require a more detailed understanding of the system being simulated and how that understanding is translated into building a simulation. Topics include: defining a detailed sequence of operations; rules of how material gets from one place to another; the interactions of people and equipment or services; and batch recipes. Templates allow logic to be implemented by describing actions rather than directly programming them. More emphasis is also placed on data input and output through exercises in building tables as well as reports for developed metrics. With the completion of this section an individual using simulation will have a much fuller understanding of the simulated environment and thus gain more from its analysis.

The final Section, for advanced users, develops skills to an even deeper understanding of the system being simulated and the simulation application. Advanced application topics include: the software architecture; implementing custom logic; scheduling operations; building new reports; and employing competing downtime events. Exercises also demonstrate how to take advantage of the application software structure. Students are able to modify simulations, making them easier to use, by creating local libraries, custom objects, and user interfaces.

A series of appendices cover some specialized topics in greater depth. These include the use of statistics, equipment reliability, interfaces for data input and output, the importance of software structure, and distributed computing.

A teacher's guide, with solutions to all exercises, and presentation materials is available from the Flexsim website. Both student and teacher forums are available for sharing experiences and tips.

Table of Contents

5. Appendix

5.1. Getting Ready to Simulate

5.2. Statistics

5.3. Interacting with Excel

5.4. Impact of software structure and language on simulation applications

5.5. Advanced Techniques

Section
1

Simulation as a Tool

Section 1-1: Working in a Global, Dynamic Environment

Economics, in the form of money, land, power, or control is the primary motivator influencing the way people live. All life throughout the world is influenced by the fact that we share the same basic elements of air, water, and land. Communications and computer technology have brought our thought processes and actions even closer together. Actions in one area now impact others as complex, and often non-intuitive, ripple effects. These ripple effects can be short or long term in nature. Even if everyone in the world believed they were working for the same objectives of peace and justice, actions by one segment could unintentionally contradict what another was doing.

Organizations, businesses, or individuals worldwide search for ways to lower costs and optimize their use of critical resources. Doing this in a changing, integrated, and dynamic worldwide environment becomes a major challenge. In recent history quality circles brought people together to avoid waste and improve their products. Today organizations search for "lean" systems that simplify efforts. They prepare value stream maps to indentify wasted time and effort. Optimization is considered a key to success. But in an ever increasing dynamic and integrated world, even these efforts can be out of date quickly or fall prey to unintended consequences.

Success in the world economy is often viewed in terms of competition, risk, and innovation. Business success involves speed and leverage in making decisions. In a time dependent world it's critical to plan, execute, and work efficiently the first time, every time. The changing demand for goods and services are just one example of the need to make timely decisions.

Simulation can directly identify areas of cost savings or efficiency as well as plying a major role in risk analysis. Simulation provides insight into risk analysis by providing increased knowledge of system dynamics and interactions. It also identifies possible scenarios resulting from various actions. Simulation may not be able to predict specific demands for future goods and services but it can be used to assess the impact of such demand variability on a system's ability to react.

All these requirements for understanding dynamic systems have placed a new emphasis on simulation technologies and applications. People across an entire organization can use simulation to make good decisions. There is now a demand for the use of simulation as a common tool to:

- **Test new concepts and options**
- **Analyze systems for improvement**
- **Match project scope with business requirements**
- **Share information across organizations**
- **Focus on the impact of complex dynamics**

To accommodate these goals, simulation has undergone a change - a change made possible by computer and software technologies. Traditionally simulation has been associated with "experts" in the field who are entrusted with providing answers and are the repository for dynamic knowledge. Such knowledge and analytical tools now need to be available to everyone as a basic tool. The result is a change in the paradigm associated with who uses simulation.

- Traditional Paradigm – Limited Base of users
  - Complex tools with no commonality
  - Experts receive training and use simulation application
  - Experts solve problems for others
  - Simulation service is expanded by adding experts

Section 1-2: Simulation in Everyday Life

Everything changes in some way, either in time, space, or through interactions. Trying to understand, analyze, and even predict these dynamics has been a desire since the beginning of time. Simulation is a tool to capture the dynamic characteristics of a system. Simulation can replicate historical dynamics, study them, and project possible future results.


Flight Simulator

Simulation is commonplace today and is often taken for granted. Simulation has been the mainstay of training systems for many years. Commercial pilots work countless hours on flight simulators before transporting the public. Almost every section of the military depends of simulators to train personnel to keep them ready for any eventuality. Operators of equipment, from supertankers to subway trains, hone their skills for handling dynamic situations that may occur.

Manufacturing has been a prime area for simulation efforts. With a single artisan working in a shop the output was very predictable. The pace was that of the individual. The only issues may have been the procurement of raw materials and the condition of the tools. With the start of the industrial revolution, and especially the assembly line, manufacturing changed into a complex entity involving multiple pieces of equipment, operations, and people --- all of which interacted with each other. Today manufacturing depends on simulation to test new manufacturing concepts, prioritize changes to existing operations, and to ensure that the outcome of a proposed project will meet expectations.

⋮

These are but a few of the many areas where simulation technology is being used – either in a very obvious way or behind the scenes. The common thread in all the cases is that simulations are created based on some level of knowledge. That knowledge may have been experience, physical laws, historical data, sequence of event tables, cause-effect relationships, or, in the case of the games, a created set of rules. The need for knowledge to build a simulation leads to the first and most important characteristic:

Simulations require either known or historical knowledge of a system. The simulation then becomes a repository of that knowledge and the resulting dynamics and makes it available to the user.

The value of a simulation, based on known knowledge, is that it represents the dynamics of the simulated system and therefore allows it to be studied and modified without disturbing the actual, physical system. Alternatively, a simulation based on created knowledge, can represent the dynamic behavior of a system that doesn't presently exist thus studying how the created rules impact system behavior.

A second characteristic is:

The simulation user is the one who gains value from the simulation

This simple statement means that those who look to others to simulate, analyze, and provide answers learn very little about the simulated operation and hence their own environment. To use and help build a simulation brings a person in direct contact with the simulated environment. In doing so a person experiences a deeper understanding of how the real world system operates and behaves.

Section 1-3: Where's 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 other 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 in various business and service sectors. The savings are in the millions.

Benefits of simulation normally fall into four categories.. Each segment has both tangible and intangible benefits associated with it. As with most characterizations there is always some overlap between the segments. The segments also line up with normal procedures for capital projects or any change to existing operations.

Section 1-4: Modeling and Simulation

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

Simulation starts with modeling and models. A model in this sense is defined as a physical or mathematical description of an object or event. It represents a single point or action in time. A simulation is then puts models in motion. An operating scale model of a piece of equipment is actually a physical simulation. Modern simulation applications are based on mathematical models but can look physical as well through three dimensional graphics.

The first form of mathematical models for fluid production facilities came in the form of equations defining the physics of material transfer. Undergraduate engineering schools taught the basics of stoichiometry starting with the material balance equation:

Input = output – change in accumulation

Processes were considered to the operating in two modes: steady state or transient. At steady state variables are not changing and the accumulation term falls to zero. The material balance equation becomes

Input = output

This level of equation, or model, was used extensively for design calculations and analysis, especially in the petroleum and chemical industries where plants would produce the same product for long periods of time. Many companies still use versions of the material balance equation in spread sheet form to model operations and calculate performance metrics. Today it's recognized that the steady state assumption is not valid over time as operations are constantly subject to changes in variables and interactions. The steady state calculation, although useful in gaining insight based on averages, is no longer sufficient to answer questions based on dynamics and statistical variations.

The term discrete event refers to those operations that have defined events associated with them. For example, in manufacturing, such events could consist of: setting up for an operation, a piece of material entering a machine, the moment the piece is worked on by the machine, and the piece leaving a machine. In a banking operation the events may be: a customer entering the bank, choosing a teller line, getting to the teller, conducting business, leaving the teller, leaving the bank.

Unfortunately, all operating systems don't nicely fall into the discrete event or fluid category. As the scope of operations to be simulated grew it became obvious that they included both fluid sections as well as discrete event sections. For example, many fluid operations contain a batching step that has an event structure but where the components being added and leaving the equipment are fluid. Operations with this characteristic were thus termed "hybrid" sytems.

Section 1-5: The Simulation Software Marketplace

Early commercial simulation programs initially focused on their specific area of expertise, either discrete event or fluid/continuous. However, because of computer and software technology at the time, they were not user-friendly and required a long learning curve. Consequently, groups of simulation specialists were formed to develop simulations and analyze the results. Many companies had large groups of simulation practitioners who focused on detailed simulations. As more individuals within the organization wanted to make use of simulation technology, they were disillusioned by the learning curve of the applications, the license costs, the time to complete a simulation, and the placement of a separate group between them and the process and problem they wanted to study.

Section 1-6: When to Use Simulation

All operations do not have to be simulated. The object of simulations is to monitor the operations being studied and obtain results. The results are most often in the form of metrics. These metrics are key variables or calculations that are needed to resolve issues or indicate performance. If the operations are in a single series without unexpected downtimes or special events, metrics can simply be calculated using a spreadsheet. However, there are several operating characteristics that should raise a red flag and indicate that the calculation of metrics is not obvious and may vary widely because of statistical variations or other events.

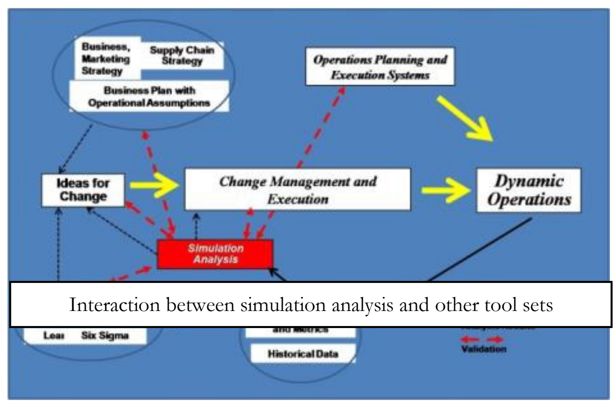
The main determining factor for deciding to simulate is the degree of operational complexity. Even the most lean and simplified operations may be operationally complex if they are influenced by external decisions, random events, or their own flexibility.

Simulated System Characteristic	Degree of Complexity
Straight through operations - Known rates - known upsets	Slight
Variables dependent on service, product or operating conditions	Slight - Moderate
Lean design with random events or external changes	Moderate
Bottlenecks with by-pass or alternative logic	Moderate
Bottlenecks that change by conditions	Moderate
Surge or KanBan areas needed	Moderate
Introduction of new operations or services	Moderate
Reliability important – Failure modes, interactions	Moderate
Interaction between fluid and discrete manufacturing areas	Moderate - High
Multiple possible material transitions or transit paths	Moderate - High
Batching Operations	Moderate - High
Scheduling influence on performance	Moderate - High
Products or services must be executed in a particular sequence	Moderate - High
Specialized setup or cleaning requirements	Moderate - High
Shared Equipment or services	Moderate - High
Coordination required with other areas or people	High
Competing down times (reliability, break or lunch times, shut downs)	High
Interaction of services, equipment and people	High
Specialized logic (e.g. dynamically adjusting variables based on condition)	High
Pull Manufacturing (make to order) operations	High

The table lists a number of conditions which can increase the degree of operational complexity. Simulation is especially useful when the degree of complexity is moderate or greater..

Section 1-7: Simulation and Other Tools

Simulation is only one of a number of tools available for efficiently studying a dynamic system. These tools extend from directing day to day operations to planning changes to meet new business opportunities. It's difficult, if not impossible to find one tool that easily satisfies the needs of users across such a wide range of activities. However, all these tools share in the objective to improve system performance as they also share in the data and metrics that drive operations. Therefore it is critical that the tools are able to share information across application, and usually, organizational, boundaries. The Figure shows the interactions between the various organizations and the tools that support the operations.



Interaction between simulation analysis and other tool sets

The Dynamic Operations block can be a production line, a plant, a service organization, or even a single operation. It is this operation that's the focus of all improvement efforts. Understanding its dynamics and metrics is the goal. Two forces directly change the operating characteristics of the system. The first is the day to day directives for operation. These may come from a direct response to economic or market conditions or from events causing the system to react. The second is the introduction of new products or services as well as changes made in hopes of improvement.

By interacting with all the various activities that impact on operations, simulation provides the important role of being a common communication and continuity tool. From idea concept to daily operation simulation provides a means for people across an organization to visualize, analyze, and discuss operations and change concepts using the same basis. Simulations of new ideas continue on from a thought to actual implementation. Along the way all individuals involved in the process see results based on the same assumptions and data. Discussions of priority or value can be compared on an equal basis. Rather than trying to discuss issues based on antidotal experience, people can base their analysis on simulated results of the actual process.

Section 1-8: Simulation Users

Not everyone who uses simulation will have the time or inclination to become proficient in all aspects of any application. Consequently users should be able to find an application that matches their needs as well as their abilities. Users should also be able to increase their simulation knowledge at their own pace. To facilitate this concept, this remainder of this book follows the progression of a user from having a casual relationship with simulation to one who is deeply involved in advanced topics. The benefits, however, are available to all levels of users.

Occasional User:

This user is a person who only uses simulation to answer specific issues or to occasionally analyze an operation. At this level, simulation is only used once or twice a year. Training must be short, the interfaces intuitive, and re-training (when simulation is again used after an extended lapse of time) minimal. Occasional users are often managers, business unit leaders, marketing personnel, engineers whose main responsibility is not simulation, or operations personnel such as team leaders. Usually these users have a run-time or limited use license for the simulation application. They are most likely the “owner” of the problem or issue to be analyzed by simulation. They have more knowledge of the target of the simulation than of the simulation application itself.

The capabilities of an occasional user and the required skills are summarized in the Tables.

Occasional User Capabilities

- Opens and Runs pre-built models
    - Validates object and system operating characteristics
    - Changes data for objects and operational scenarios
    - Uses pre-built reports to analyze metrics
- Constructs visual Dynamic Functional Diagram (DFD)
    - Populates simulation screen with objects
    - Makes basic input and output connections
    - Changes basic object characteristics

Occasional User Capabilities

- Basic understanding of simulation
- Opening pre-built simulation
    - Three dimensional screen navigation
    - Use of menus
    - Changing object and operating scenario values
    - Analyzing simulation output and object statistics
- Developing a DFD
    - Understanding of libraries and the objects in them
    - Moving objects to the simulation surface
    - Making changes to object data
    - Adding and removing object connections

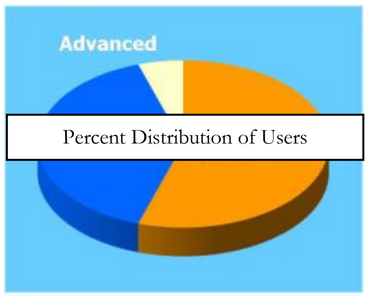
Intermediate User:

The Intermediate User is an active simulation practitioner. This user usually has some problem solving role as a primary focus such as an engineer in a corporate or operations group. Intermediate users are also found in logistics, research, and business unit functions. For this user, simulation is a tool that is available to help with the person's main responsibility.

Advanced User:

Building on the skills of both the occasional and intermediate user, the advanced user can build complex models with extensive custom logic. While some organizations may devote a person to the simulation task, the primary responsibility is usually still not simulation but some internal consultant function. This user will be called on to help simulate and analyze conditions or systems that are beyond the scope of other users. This person most likely is the primary technical contact with the simulation provider.

Organizations making use of simulation may have anywhere from one to a number of people with various levels of simulation skills to aide in the responsibilities. Large organizations may have a distribution of users shown in Figure. Ideally the vast majority of users have occasional and intermediate user skills. Modern simulation applications make it possible to easily obtain those levels. Since all levels are often needed at some point to gain the most out of simulation, organizations with only a single occasional user will often outsource the advanced skills to the software vendor or third party consultants or engineering firms.



Percent Distribution of Users

However, the value of simulation will only be a achieved if an organization maintains at least a person or persons at the occasional user level or above.

Section 2

The Occasional User

Section 2-1: Using a Simulation

An Occasional User is one who doesn't use simulation on a regular basis but understands the principles of simulation and 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 simulations but, with a little review, can make us of a simulation built by others – especially if the simulation has intuitive interfaces supplied with the application. It is assumed that the user is familiar with the subject matter of the simulation, however.

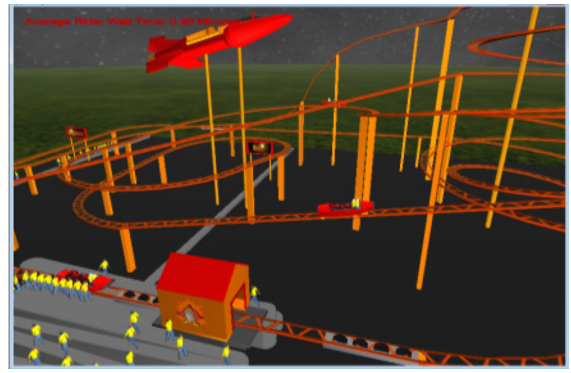
The occasional user knows enough about simulation to create a functional specification for others to build a simulation based on the user's knowledge of the system being simulated. As the "owner" of the problem or issue to be studied by the simulation, the occasional user can detail the simulation requirements, define the scope, and establish the metrics for the analysis.

Through exercises, the first part of this section will develop the skills and knowledge base for the occasional user. Each example contains background information, a statement of the problem or issue, information about the system being simulated, and the expected results. Where required, information needed for using the simulation software is included. This information is a quick summary of how Flexsim is used for the particular example. The help, Getting Started, and Tutorial sections of Flexsim provide an excellent way to learn more in-depth techniques.

Exercise 2-1-1: Coasting Around

Background:

As the manager of the "Super Rocket" roller coaster ride at the Nine Flags amusement park, you 're 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 have to show improved financials or find another job. They're a lot of trade-off 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. 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 involved costs of equipment or staffing.

Your brother-in-law, whom you normally can't stand, happens to be familiar with simulation methods and offered to help. You explained your problem and provided base data about the operation. While appreciative, you really didn't believe he could make the decisions based on the simulation so you decided to run the simulation and analyze the results yourself.

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

Coaster Operating Data:

> ***Preview Note:***
>
> The "Operating Data" part of the exercise contains specific data needed for the exercise

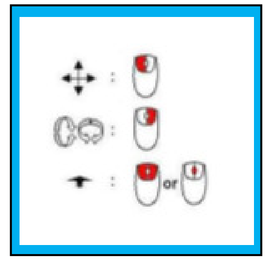
Expected Results: Provide a plan for operation that includes staffing. Determine if there is sufficient return to request the addition of more car(s).

What You Need to Know:

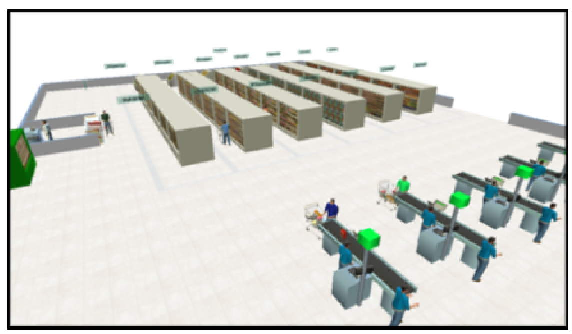
Open the Flexsim software by double clicking on the desktop icon – If necessary, check Appendix 1 for instructions to obtain and load the Flexsim student software. Choose to open a saved model and navigate to the Text book Example folder and select Section 2-2. After loading, the simulation will open as shown in the figure. This is the model view window. On the right side of the screen is the control Graphical User Interface (GUI) that can be used to operate the model, change variables, and view results. In subsequent examples, other methods will be used to achieve the same functionality.

> ***Preview Note:***
>
> The "What You Need to Know" part of the exercise contains the necessary information for the user to successfully complete the exercise

The Flexsim visual environment is three dimensional and most Flexsim navigation will be done using the mouse. It is recommended that a mouse with both left and right buttons and a scroll wheel be used for maximum efficiency. Executing a left-click and drag in an empty area of a view window, you may translate the view left/right, up/down and diagonally. You may tilt and rotate the view by right-click and dragging. This will manipulate the camera angle looking into your model. If your mouse has a scroll wheel, scrolling up will zoom in on your model and scrolling down zooms the view out. If the mouse doesn't have a scroll wheel, the same action can be accomplished by holding down both mouse buttons and pushing the mouse forward (zoom in) and backward (zoom out). These moves are summarized in Figure. If, while manipulating a view window you end up not liking the results or even getting "lost" in the view, it can be reset by right clicking on the view, choosing "view", and then the "Reset View" option.



Exercise 2-1-2: Farm Pride



Background:

As regional manager of the ***Farm Pride*** chain of local supermarkets you are considering expanding with a new market in the Snyder's Point area. You've found an ideal location that should attract working families who favor specialty foods along with staples and prepared meals. The contractor wants to know design details and your HR department wants to know staffing levels. You've asked your business development group to assist in the design. Based on other store designs they have a simulation to help with the design but you have to perform the analysis yourself.

Problem Statement:
Based on general trends of shoppers in the proposed market area, what staffing levels should you anticipate? Identify general rules for the number of registers open based on the number of people entering the store.

Operating Data:

Expected Results:
Prepare a report that will identify staffing levels for various times of day and week, what service levels are needed at the deli counter, and if employees can share work tasks.

Exercise 2-1-3: Slime Inc.

> ***Preview Note:***
>
> The additional exercises in this section are designed to guide the user through additional skills associated with opening and using the pre-built simulation.

Exercise 2-1-4: Department of Transportation

Exercise 2-1-5: Grandma's Pie Emporium

Exercise 2-1-6: Martian Transfer Station

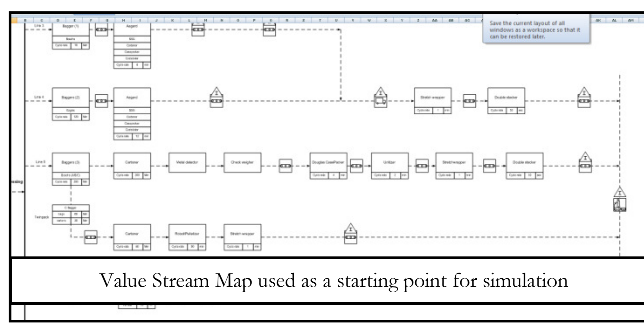
Exercise 2-1-7: Default Motors

Exercise 2-1-8: Sandy Lee Snacks

Exercise 2-1-9: Homeland Security

Exercise 2-1-10: Brandon's Flour

Section 2-2: Specifying a Simulation



Value Stream Map used as a starting point for simulation

If a simulation is not available and the Occasional user would like to utilize simulation for analysis, the user may describe the system for a more advanced user to simulate. The functional description for a simulation requires more information than a standard block diagram. A value stream map such as shown in the Figure can be a starting point as considerable data has already been compiled. The value stream map, however, is not sufficient by itself for defining a simulation as it typically lacks the logic used to make material transfer decisions or detail the sequence of operations.

The functional specification for a simulation is a joint effort between the person who wants the simulation and the person developing the simulation. The result will only have real value to those people involved with the work. Defining a simulation, in a way to assure that the resulting model will be of value, is straightforward. It requires answering the following four questions.

Section 2-3: Documenting the Specification

Documentation of the functional specifications takes two forms – a written document and a simulation layout. The Simulation Project Template is a vehicle to convey the basic simulation description as well as serve as a continuity document covering the development of the simulation as well as documentation for future users. The further away the simulation "owner" is from the simulation developer, the more important is the written specification. The template is available as part of this book.

The documentation template consists of three sections. The first documents the functional specification; the second documents the way the developer built the simulation and how it is operated; and the final section is a check list establishing that the simulation meets the functional specification.

Section 2-4: Creating a Dynamic Functional Diagram (DFD)

The second part of defining a simulation is the creation of the Dynamic Functional Diagram (DFD) in the simulation window. At a cursory level the diagram looks similar to a value stream map or block diagram with visualizations of equipment instead of boxes. It especially starts out that way when created by an Occasional User. As the simulation is developed, however, the visual imagery and dynamic logic that is added takes it far beyond the usefulness of other diagrams. The DFD, like the simulation project template, is a continuity document and continues with the simulation activity. More importantly, the simulation is built as a continuation of the DFD.

Exercise 2-4-1: Plant Expansion Project

Background: As the primary engineer working for a company making personal care products, you are responsible for successfully commissioning a new production line. The plant already has five lines, each producing a different product. Cases of product from each line accumulate in groups waiting to be palletized. The cases proceed as a group across a shared conveyor to a single unitizer that loads the group of cases onto a pallet. The pallet is picked up by a fork truck and moved to the warehouse. You are worried that the new line will exceed the unitizer capacity although a check of the average production for all the lines would indicate that capacity still exists. The engineering firm handling the installation of the new line indicated that they could create a simulation for use in establishing the unitizer capacity.

Problem Statement: Can the new line be successfully added to the plant without any loss in production?

Operating Data:
Value Stream map
Process data
Historical operating data

Expected Results: Prepare a functional specification using the first section of the simulation project template. Create a simple DFD focusing on the area chosen for the simulation.

What you need to know:

Building a DFD starts by opening the Flexsim software and choosing to build a new simulation. The Figure shows the default window configuration. The main sections of the Flexsim interface have been marked off with the numbers. Section 1 is the Flexsim Toolbar, which, in addition to containing the familiar Windows style set of menu options, also has buttons that allow quick access to some common Flexsim interface elements. The buttons include creating a new model, opening a previously saved model or saving the current model.

In the next button grouping one may select the object connection/disconnection mode for the cursor. Access is also provided for the advanced model views on the toolbar, such as the tree view, orthographic view and the perspective view. The orthographic view is the default and is usually best used when setting up and connecting objects. The Perspective view gives a representation that shows depth and creates a somewhat more realistic view.

Preview Note:

The exercises in this section help to build the user's knowledge of evaluating various systems, focusing on the key elements to simulate, and accurately defining the scope and requirements for the simulation.

Exercise 2-4-2: Harvey Confections

⋮

Exercise 2-4-3: Burgers_R_Us

⋮

Exercise 2-4-4: Molson Dairy

⋮

Section

3

The Intermediate User

The Intermediate User is a person who uses simulation as part of their job description. Their main focus, however, is other than simulation. Such job descriptions include a process, packing, or manufacturing engineer. Individuals engaged in manufacturing research and development may also be intermediate users. During a year, the Intermediate user will use simulation 5 or 6 times. Intermediate users can build relatively straight forward simulations using drop-down menu or wizards. When more detained simulations are required, the intermediate user can call on technical support or third-party consultants.

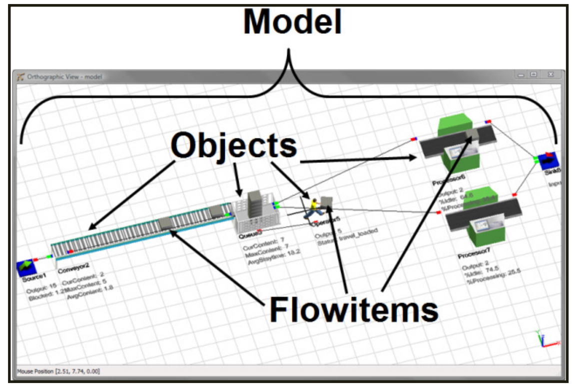
The remainder of this section contains exercises that will develop the intermediate users' skills. The first part introduces basic simulation building principles and details for using the Flexsim product. The discussion is meant as a quick start and minimum detail for completing the exercises. Additional summary details are included with the exercises as needed. The imbedded manual in the "help" sectin of the Flexsim application should be used for more details.

Section 3-1: Learning the Basics

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 are basically the same.

Models, Objects and Flowitems: A Flexsim **object**, is the most basic building block of a simulation. They are found in the Library and can be brought out to the model workspace as used in the previous chapter. A **flowitem** refers to the actual discrete entities that flow through a simulation. Depending on the simulation, these would be products, customers, paperwork, and so on. A **model** is simply a collection of these objects (flowitems and Flexsim objects), put together in such a manner as to simulate a system.

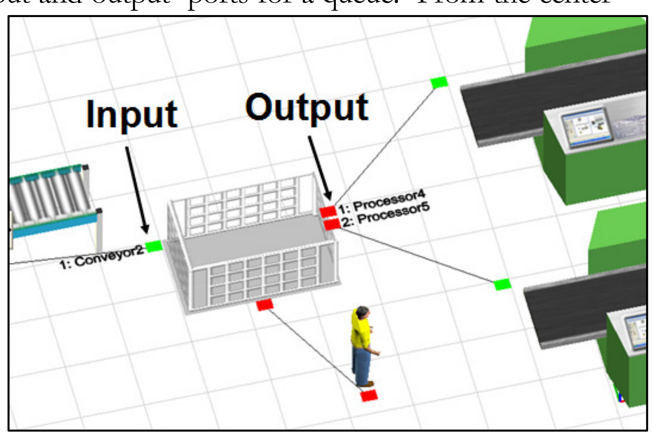
The **Flexsim Model** in the Figure is a combination of Flexsim objects (queues, processes and transportation) and flow items. Opening the Flexsim application, starting a new model, and dragging out and connecting the objects as shown in the Figure is a good way to follow the discussion. If you make the connections, click on the Reset and Run buttons, flow items will appear based on default values.



Section 3-1-1: Making Connections

Fixed resources move flowitems through models using **port connections**. These port connections establish the relationships needed between objects to define a **flow**. To establish an **output port** to **input port** relationship, connect the output port on an origin object to the input port of a downstream object. Each output-to-input relationship defines a possible routing between fixed resources. Any possible routing choice in the model needs to have a connection that defines it.

The Figure shows the input and output ports for a queue. From the center of the queue is another connection to a task executer that's in the form of an operator. This connection is called a **Center Port** connection. A center port relationship is a way for one object to reference and communicate with another. Since objects in the model aren't aware of each other, if one object needs to communicate with another some kind of reference needs to be established first. The center port allows this to happen. One of the more typical uses for a center port connection is as depicted in the figure. Other ways of communicating between objects will be discussed later.



Section 3-1-2: Editing Objects

Open Flexsim and drag out a processor object to follow with the text

All objects may be edited by first double clicking on the object. The editable set of attributes is referred to as the **Properties.** The properties of an object deal with what the object does and how it behaves. For example, how long flowitems will stay inside the object, where they will go when they are released, as well as how it looks.

Each object class has its own set of properties unique to that class, though they will share a similar tabbed layout. Attributes in the same tab are ones that relate to each other. Nearly all objects will have a tab named **Flow**, which sets the rules for how flowitems move and a tab named **Triggers,** that allows the user to add additional functionality as a response to events on an object.

Exercise 3-2: The Box Office

Background: The Harrington Center was originally built as a multipurpose venue for sporting and entertainment events. It never reached its potential, however, and was used mainly for the local community college's basketball games. Kantor Promotions felt the building had promise and purchased the Harrington Center with plans to expand its use. The facility had a single box office window. Customer satisfaction will be critical to success especially when selling tickets

Problem Statement: How should the box office be set up and operated.

Operating Data: Ticket sales demand will vary considerably depending on the number of events at the venue and the time of day. A starting point will be based on historical sales during the two hours before an event. At that time customers arrived every 60 seconds on average with the actual arrival time best described by an exponential distribution with a location value of 0 and a scale value of 60. The time to actually purchase the ticket varies with a lognormal(31,31,0.5) distribution. Observations have shown that if the ticket line exceeds 20 people, even the best fans will leave and listen to the game on the radio.

Expected Results: Validate the model**.** Determine how many people you turn away during the operating day – max, min average.

Preview Note:

The "What You Need to Know" section contains information specific to completing this exercise

What You Need to Know: This exercise will start off with a simple model. It's always a good idea when building a simulation to make sure the logic and flow are correct. Open Flexsim and select to make a new model. Once open it often is a good idea to open the view settings tool bar – under the "view" drop down menu, select "modeling utilities" and then ""view settings". The tool bar can moved anywhere on the screen. This tool bar allows the model view screen to be changed by turning on/off the grid, connections, names, etc.

Exercise 3-3: Box Office Expansion

Background: Kantor Promotions is willing to spend money to bring in more attractive events but want to be sure that, when needed, they will meet the demands for tickets. If the Harrington Center staff can't come up with a viable plan they will outsource the majority of ticket sales to Ticket Master.

Problem Statement: Find a plan for additional ticket windows and/or ticket sales that would maximize profit .

Operating Data:

Expected Results: Validate the model. Determine how many people you turn away during the operating day – max, min average.

Preview Note:

The remaining exercises in this section each focus on a simulation functionality such as: conveyors, pipes, tables, changing images, queues, logic triggers, flowitems, basic reliability, using operators, transportation, statistical distributions, and interfacing with Excel.

What You Need to Know: There are many ways of creating the simulation for the second exercise. In keeping with the Intermediate user, the following steps will not require any direct programming. Until the simulation is tested and ready to run, turn off the Experimenter from the last Exercise by choosing the experimenter from the Tools drop down menu and un-checking the box that says "Use Experimenter". Use "save as" in the file drop down menu to keep this model as a new name. Save work often.

⋮

Exercise 3-4: Keggler's Brew

⋮

Exercise 3-5: Bio Fuels , Inc

⋮

Exercise 3-6: Johnson Pharmaceuticals

⋮

Exercise 3-7: Hamilton Airport

Exercise 3-8: Standard Toys

Exercise 3-9: Beman's Bakery

Exercise 3-10: Audio Services

Exercise 3-11: Jelly and Jams

Section

4

The Advanced User

While the Intermediate user can study a wide variety of dynamic systems using simulation, there will still be a need to simulate actions that can't be described using the preset logic. The advanced user has enough understanding of the simulation application to create custom logic and use advanced features to analyze more complex situations. The ease of accomplishing these tasks means that the advanced user need not be a simulation or programming professional. Most advanced users will either have a definite interest in simulating systems or carry out simulations on a relatively frequent basis. The 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 application, it does not completely describe all the advanced functionality available.

The primary background required for the Advanced user is a knowledge of the underlying software structure and the basic commands for building custom logic. The Advanced user also becomes more proficient in dealing with information that is exchanged between the simulation and external programs. Additional skills include utilizing the extensive charting and reporting functionality of the application and the ability to enhance the visualization process by creating additional dynamic action paths. This section provides an overview of the software architecture and scripting commands. More information is provided in the Flexsim user manual under the "Help" drop down menu.

Preview Note:

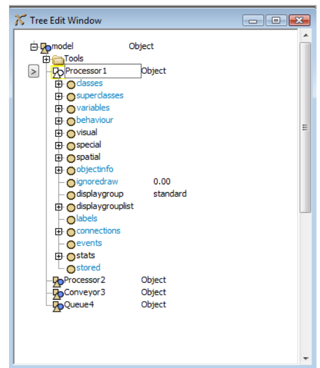
These first sections prepare the Advanced User with the general skills to begin to use more advanced simulation features including customized logic.

Section 4-1: Software Architecture - The Tree Structure

In a general sense the Tree is a way to arrange data such that data points in the tree have a hierarchical relationship to each other. These data points are called **Nodes** and are the most basic data structure in Flexsim. The nodes in the Flexsim tree hold all the information that a Flexsim model uses to run a simulation. All the parameters and properties for an object, graphical user interfaces, the event list, and so on all are stored in the Flexsim tree. The structure of the tree is very much like a typical computer hierarchical file structure.

The tree provides a very robust and efficient way to find where variables are stored or where particular sections of code reside. It is a modern alternative to searching through lines of "spaghetti code" and confusing subroutines to find a reference to a variable or to know where certain commands are used. It's a feature unique to the Flexsim application and makes addition custom logic straightforward.

To see the tree view of a model, open a new model space and drag out a few objects such as a processor, conveyor, etc. In a blank space on the surface right click and select "explore tree" from the popup menu. The tree structure appears with the "model" as the first reference as shown in the Figure. Holding and moving the left mouse button scrolls through the tree as does the scroll bar on the right. Left clicking on an individual object in the simulation and choosing "explore tree" will open the tree structure for that object.



Section 4-2: Scripting Basics

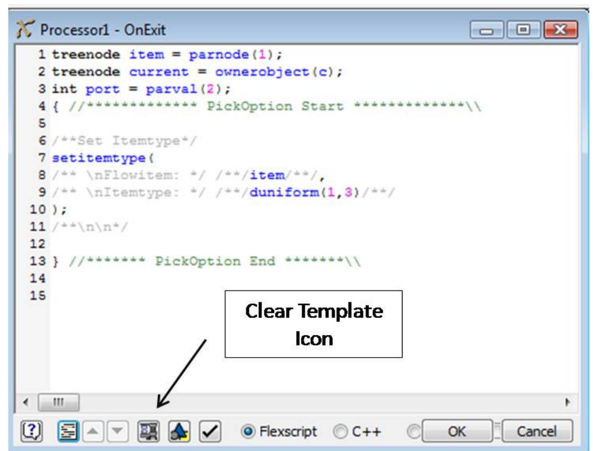
The picklist options, introduced for the Intermediate user, provide a wide range of commands to control the simulation without writing any code. However, if simulation is intended to examine truly unique strategies that provide a definitive advantage, then writing some level of unique, custom code is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

Flexscript was modeled around the syntax for the popular programming languages of C/C++ and follows many of the same conventions found in those languages. Anyone familiar with C/C++ at all can quickly begin writing custom modeling code. Some basic rules and guidelines:

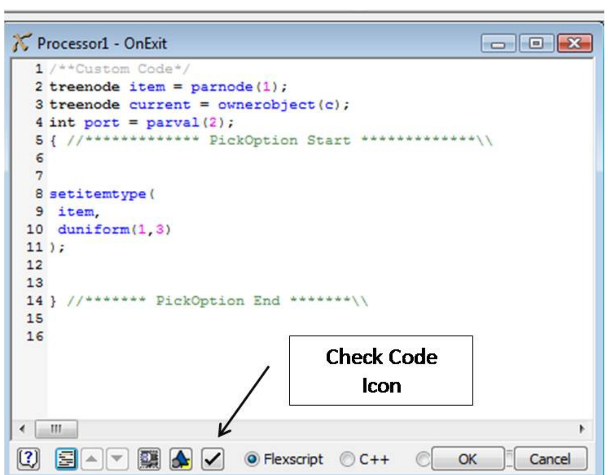
- The language is case sensitive. "A" does not equal "a", "Processor1" is not the same as "processor1".
- No specific formatting is required. Use of spaces, tabs and line returns is encouraged for more "readable" code.

Section 4-3: Creating Custom Logic

Flexscript code can be added nearly anywhere within the simulation. For this text, however, most custom logic will be added in pick options. The script window is found by clicking on the "A" icon associated with a pick option.



The Figure is the view of Exit trigger of a processor after the "A" icon has been pressed and where the template to set the item number has been selected. At the top of the window three variables have been defined by default. These include the key words ***item*** and ***current*** as well as the ***port***. The use of comment fields and other commands are needed for the code template view. To see the code without the additional comment fields and to prepare for adding or modifying the code, click the "clear template view" icon at the bottom of the window as shown in the Figure.



The resulting window starts with a comment section containing the words "custom code". Replacing this string will change the text in the drop down window that defines the logic. There remain comment sections that indicate the range of the pick option – in this case changing the color of the item in the processor. Another important icon in the window is the checkmark at the bottom of the screen. Clicking on that icon checks the syntax of the code. Any errors will pop up in a compiler consol window at the bottom of the Flexsim window. The consol will indicate the type of error and the line number.

There are visualizations to help keep the code correct.

⋮

Exercise 4-4: Purity Soap

Background: Consumers have always liked being able to purchase laundry soaps in various size containers. Recent increases in manufacturing costs and the reluctance of customers to pay higher prices has forced marketing to rethink their size/price mix. As a result, the plant has been forced to run many sizes of product with the associated change over times. Since the plant is rewarded on the total number of cases produced, the change in product schedule is not always optimal. You've been meaning to simulate the plant operations and this seemed like a good opportunity to please the plant manager.

Problem Statement: How can the plant respond to the demands of the market so that commercial and plant goals are met.

Operating Data: The plant runs three lines that can produce 10 SKUs. The associated spreadsheet has all the operating data of package sizes, rates, and potential schedule requested by marketing.

Preview Note:

The "What You Need to Know" section contains information specific to completing this exercise

Expected Results: A meeting is scheduled with the laundry soap business unit. Build a case to discuss possible guidelines for production schedules, modification of size types, or changing production goals.

What You Need to Know: Operating multiple production or packing lines to meet a schedule requires assigning products to run on each line and controlling the actions of all equipment on the line. The equipment for the line must be started at a particular time, all operating variables for the particular product established on each machine, a tally kept for when to stop the line at the appropriate target, cleaning out the line, and changes made in preparation for the next production run. These function can be performed through the use of specialized system and line controllers.

Exercise 4-5: James Chips

Background: James Chips is a family owned and run business which has seen significant growth over the last few years. After producing a home-made, gourmet potato chip for many years, James Sr. finally gave in to his advisors who recommended expanding their line with "flanker" products – namely flavored chips.

The small manufacturing group came up with a design that they felt was flexible and lean. It involved a batch makeup system of flavoring to add to the chips. Regular chips could be run on the line while the flavor system was cleaned for the next product. Everything went well until production started increasing and the flavor system became the bottleneck. The group couldn't understand the problem since their calculations showed the batch make up time was short enough to handle the higher rates. As a second cousin and on the manufacturing team, you decided to take a different approach to the problem.

Problem Statement: What will be required to remove the bottleneck at the higher production rates.

Operating Data: The data for the example is in the spreadsheet and includes rates, logic, and flavor batching recipes.

Expected Results: Show what changes in the process are needed and calculate the resulting rates.

What You Need to Know: Simulate the operation to confirm the bottleneck. Observe and analyze the results to determine any corrections. The flavor batching has a step that requires an operator to add bags of flavoring to the mix.

Preview Note:

The remaining exercises in this section each focus on advanced simulation functionality such as: production schedules, Global variables. Excel interaction, User interfaces for data entry and results, Batch makeup, pulling items, team tasks, etc.

Exercise 4-6: Hilltop Steel Works

⋮

Exercise 4-7: Newland Bay Ferry Service

⋮

Exercise 4-8: Canton Supply

⋮

Exercise 4-9: Pineville Custom Cabinets

⋮

Exercise 4-10: Emergency response

⋮

Exercise 4-11: License Office

Exercise 4-12: Swift Lunch

Exercise 4-13: Apex Distribution Center

Section

5

Appendix

1. Getting Ready to Simulate

The first step required to follow the exercises in this text is to download and load the current version of the Flexsim Student Edition. The first page of the text contains a serial number that is used for registering the software at www.flexsim.com. If this book is being used in conjunction with a registered course, extra libraries and media files that support the text simulations will be included.

2. Statistics

Statistics play a critical role in all aspects of simulation. Statistical distributions are used to represent input events, cycle times, down times, and other functions that occur in a probabilistic manner. Statistics also are used to analyze and report on simulation outputs. Neither this book nor this appendix section is intended to be a course on the use of statistics but rather explains how to use the robust statistical capability of the application.

A complete set of statistical commands with examples can be found in the "Help" section of the application. This appendix serves to summarize both the input distributions and the output analysis options.

During a simulation, the application automatically tracks many different data points involving all simulation objects. These data points may be recorded in the charting software to create a data base for generating graphs and reports using the simulation results. Functions include tracking the movement of material through the system, performing a financial analysis, or comparing the statistics of various objects.

3. Interacting with Excel and other applications

While the exercises in the text exchange input and export information from Excel, these actions are accomplished through a template format that does not need the user to write any scripting code. However, the Flexsim has a full complement of commands for creating custom interfaces with Excel and other applications.

4. Impact of software structure and language on simulation applications

The functionality of simulation applications is directly tied to its software language and underlying architecture. Capabilities such as visualization, ease of use, and compatibility with other applications require a modern structure. The Flexsim software structure allows it easily to facilitate Virtual Reality modeling, design of experiments, result analysis, data collection, system optimization and user creation of applications complete with powerful user interfaces. Flexsim supports C++, OpenGL and provides its own language scripting capability. Flexsim is architected as an extensible, modular and open (or closed) source platform.

Flexsim is developed using object oriented methods and C++, with a proprietary database abstraction layer. It is also interoperable with external SQL databases. Flexsim runs on Microsoft Windows and has a bi-directional interface to Microsoft Visual Studio. Flexsim's distributed simulation application has an extra network layer facilitating managed data transfer between Flexsim stations running on different computers and/or on the same computer. This architecture provides a secure, robust and extensible system for managing compute intensive simulation studies locally or across a network.

⋮

5. Advanced Techniques

While this text covers a wide range of operational systems that can be analyzed with simulation, it doesn't cover the wide range of functionality included in the Flexsim software. This appendix summarizes some of those additional capabilities such as creating custom task sequences for transporters or other objects to follow, custom user interfaces to control the simulations, and visualization techniques such as advanced animation and adding a CAD drawing to the base of a simulation. More detailed explanations and examples are found in the Flexsim "Help" tab.

Task Sequences: A task sequence is a series of tasks to be executed in sequential order by a Task Executer. Task Executers includes Operators, Transporters, Cranes, ASRS vehicles, Robots, Elevators, and other mobile resource objects. Fixed Resources, such as processors, have a default mechanism for creating task sequences to move flow items to the next station. They also automatically crate a task for calling an operator during certain parts of the operation or on breakdowns. These default mechanisms were used in the exercises in the text. However, there are times when tasks are needed that have not been pre-defined.

⋮

About the Authors

Malcolm Beaverstock, PhD. 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 lead 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.

Eamonn Lavery, PhD. Dr. Lavery completed his first object-oriented simulation software while in graduate school. In 1999 Dr. Lavery 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 Dr. Lavery Joined Flexsim Software Products, Inc. as the Chief Technology Officer. Dr. Lavery is the author and architect of the Flexsim Software.

Dr. Lavery 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 Mr. Nordgren founded ProModel Corporation and was Vice President until he left in 1992. In 1993 Mr. Nordgren founded F&H Simulations, Inc. (now Flexsim Software Products, Inc.) and introduced Taylor II, Taylor ED, and Flexsim into the market. Mr. Nordgren has authored several papers dealing with simulation project management, queuing theory, and has taught hundreds of classes in the use of simulation software. Mr. Nordgren is listed in Marquis Who's Who in America for his accomplishments in the advancement of simulation technology. Mr. Nordgren received a Bachelor of Science in Manufacturing Engineering Technology, and a Master of Science in CIM (Computer Integrated Manufacturing) from Brigham Young University.