5 Simulation Quality and Development

Building the model is often the step in the simulation life cycle analysts anticipate eagerly. The process begins with selecting a tool or language for development and ends with a complete model, ready for experimentation and output analysis.

5.1 Quality Assurance Phase

Quality assurance is an ongoing activity in simulation model development. That being said, several activities are keys to ensuring a high quality simulation emerges during development. Among these are validation concerns.

5.1.1 Validation

Validation is used to determine the real-world system being studied is accurately represented by the simulation model. In other words, this process ensures the conceptual model is correct and establishes an acceptable level of confidence the conclusions drawn from running the simulation will give insight as to the true operating characteristics of the system being modeled. This confidence, called face validity, should radiate from both modeler and model user.

Validation is ongoing and should be part of the simulation process from the very beginning and continue until the very end. Validation techniques are particularly important during the phases of the simulation process where input data are collected, analyzed and structured for use in the model.

5.1.2 Simulation Input Data Validation

As stated in an earlier section, simulation inputs are gathered in two formats, qualitative and quantitative. Qualitative inputs are the underlying assumptions, rules, and other non-numeric data that will be used in the simulation. This information should be gathered and validated through one or several of the following methods:

1) Observation: If a model of an existing system is being developed, the analyst can observe different situations and ensure that the assumptions to be used in the model are valid.

2) Expert's Opinions: A list of assumptions and rules can be evaluated by experts. A modeler should interact with both system experts and model users throughout the life of a simulation project beginning with development of the input data.

3) Intuition and Experience: If a simulation analyst frequently models systems that share many common characteristics, he/she will develop an intuitive feeling that will help give the model added validity.

Quantitative or numeric inputs for a simulation should be tested for validity in the following ways:

1) Statistical Testing: If a theoretical input data distribution is being used to model empirical data, chi-square, Kolmogorov-Smirnov, or other goodness-of-fit tests should be used to assess that data. If the fit is close, the theoretical data can reliably be used as a valid representation.

2) Sensitivity Analysis: Involves altering the model's input by a small amount and checking the corresponding effect on the model's output. If the output varies widely for a small change in an input parameter, then that input parameter may need to be re-evaluated. Highly sensitive model behavior may indicate problems with the coding or input data.

5.1.3 Validation of Simulation Outputs

Often, the best test of model validity will not come until later in the simulation life cycle. Most of the time, the best test is to conduct a thorough analysis of the simulation's output data. If the model's output data closely represents the expected values for the system's real-world data, then validity is more likely.

When a model has been developed for an existing system, a validity test becomes a statistical comparison. Data collected from actual system operation can be used as a theoretical comparator. When the system does not yet exist, validity becomes harder to prove. In many cases, validity cannot be definitely proven until some point in the future when the system being modeled has been installed and is operational. Before that time, several methods can be used to increase the confidence level that a model is valid:

1. Comparison with data from a similar system: If a system exists which is similar in nature to the one which is being modeled, an interpolation of system output can be derived and compared to the simulation.
2. Expert opinions: An expert on the type of system being modeled can be consulted and shown the output data. His/her opinion will help lend more confidence that the model is valid.
3. Calculated expectations: In some cases, expected output can be calculated and the model output compared to the result. If the model is too complex, it may be possible to analyze individual subsystems and do calculations on each part to help establish validity.

5.1.4 Summary of Validation

Validity is an important aspect of simulation. Without confidence that a simulation is valid, model output is worthless. One key to validity is communication. With continual interaction between system designers, component vendors, experts, model users and system analysts, the perceived and actual validity of a model will be greater.

5.2 Selection of a Language or Tool

Following completion of a concept model, a decision must be made to halt the effort and be content with currently available information or proceed with development of a detailed simulation. If further development is the chosen course of action, then an appropriate simulation language or tool must be made.

As mentioned in an earlier chapter, simulation software products can be broken into two major categories: simulators and simulation languages. Simulators are application specific prewritten modeling tools. Simulation languages, on the other hand, are general purpose programming languages specially adapted for modeling applications.

Choosing a software product can be a difficult process. Software is an intangible entity with numerous parameters and features to compare. Many vendors, based on their own testing criteria and definition, claim to have the fastest, most user-friendly, or best package available. While it may not be possible to select an "absolute best" simulation package, it is very possible to select a simulation package that will meet the requirements of a particular project.

Software evaluation can be broken into four steps (Figure 5.1):

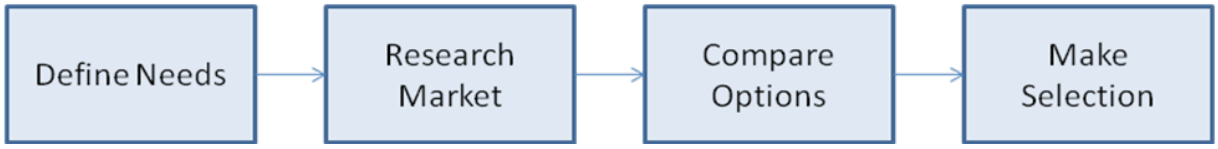

Figure 5.1 Software Evaluation Activities

Step one: Define needs – This is an information gathering step during which the prospective software purchaser needs to answer several questions.

1. Determine the frequency of future simulation work. Will this be a onetime project or will simulation be used regularly from this point on?

   If a simulation is to be done only once for this project, the amount of time spent on the learning curve should be minimized. This can be accomplished by purchasing a user-friendly simulator or by hiring a simulation consultant. If simulation work is to be done on a regular basis in the future, then serious development of an in-house simulation capability is desirable. This development would involve the training of appropriate personnel and the selection of either a general purpose simulation language or appropriate simulator.

2. What type of system will be modeled? Is it unique in function?

   If the system to be simulated is unique, then it will be hard to find a simulator package that can be used without trouble. In this case a general purpose simulation language should be used. If the system being modeled is standard materials handling equipment being purchased for in-house use, the probability that a simulator is available is quite high.

3. Who will be doing the simulation work?

   If an engineer who is not a programmer and not a computer expert will be doing the work, a simulator package may be desirable. On the other hand, a simulation analyst might feel constrained by a simulator package and might desire the full range of capabilities a language offers.

4. What budgetary constraints exist?

   The purchase of all tools and equipment is dependent upon a budget. Simulation software products exist in nearly all price ranges. Budget will have an impact on what can be bought.

5. Who will be using the results?

   Will this simulation study be used to "sell" an idea to management? How impressive does the output need to be? If it is going to be used as a research tool, then good statistical output is desirable. If it will be used to impress a potential customer or upper management, then a top notch animation might be desirable.

Answers to the preceding five questions will help start the software evaluation process by giving definition to the general needs of the prospective buyer.

Step Two: Research Market – A large and varied simulation software market exists. Based on the needs analysis conducted in Step One, a list of important software features can be generated. Products exhibiting some or all of these features can then be researched further.

Many simulation software vendors advertise their products in trade magazines popular in a particular target market. Others provide extensive websites with examples of use, free downloads of trial software versions, and successful case studies.

Another source of information about simulation software is the Winter Simulation Conference. Its Website is http://wintersim.org/.

Step Three: Compare Available Options – The process of selecting simulation software is much like shopping for any other commodity. Certain features are more important than others. No one product will be able to provide the best of each desired attribute, so a somewhat subjective decision will have to be made.

In order to facilitate the process of comparing different software packages, a list of important attributes needs to be generated. Each attribute in this list should be given a weighting factor. In this way an evaluation template is created. An evaluation template can be easily creating using spreadsheet software. The following example illustrates (Table 5.1):

	Weighting Factor		Product Score		Total
Graphics	3	X	_____	=	_____
User-Friendliness	8	X	_____	=	_____
Software capability	5	X	_____	=	_____
Ease of Use	9	X	_____	=	_____
Power	4	X	_____	=	_____
Cost	10	X	_____	=	_____
Output reports	6	X	_____	=	_____
Ease of Debugging	8	X	_____	=	_____
Statistics	7	X	_____	=	_____
Customer Support	8	X	_____	=	_____
Training	8	X	_____	=	_____
Documentation	8	X	_____	=	_____
Vendor stability	2	X	_____	=	_____
Application specific modules	9	X	_____	=	_____
			Total Product Score =		_____

Table 5.1 Simulation Software Evaluation Template

T-Co's Search for Simulation Software

T-Co is a manufacturer of t-brackets, hinges, and other small hardware items. They plan to install a new series of CNC machines to increase their capacity. Before making a major investment, management at T-Co decided to simulate the proposed system. They identified the following attributes as being important in the simulation software selection process:

- This will be a onetime simulation project.
- The system being modeled is not unusual or unique.
- An in-house engineer will be doing the simulation work.
- An inexpensive package should used.
- Results will be for in-house analysis. Animation is not important, but would be nice.

The engineer assigned to the simulation project at T-Co developed the evaluation template in Table 5.1 to help in the software selection process. The weighting factors (1 to 10, with 10 being most desirable) were subjectively developed from the attributes identified by management. The weighting factor is multiplied by the product score to produce a column of totals. These totals are added to give an overall numeric product score.
Before the engineer was able to really get started on his evaluation process, the management at T-Co decided simulation could be used for other future in-house projects. They went through their requirement list and altered it to read as follows:

- Simulation will be used on a regular basis at T-Co.
- The systems being modeled are not unusual or unique.
- An in-house engineer will be doing the simulation work.
- The best package, without regard to cost, should be used.
- Results will be used both for in-house analysis and as a tool to sell our ideas to our parent company. Animation can be used to help perform this function.

The project's engineer received the new requirement list and modified his template as shown in Table 5.2.

Table 5.2 Modified Simulation Software Evaluation Template

	Weighting Factor		Product Score	Total
Graphics	10	X	_____	= _____
User-Friendliness	6	X	_____	= _____
Software capability	10	X	_____	= _____
Ease of Use	5	X	_____	= _____
Power	9	X	_____	= _____
Cost	2	X		=

Table 5.2 Modified Simulation Software Evaluation Template

Step 4: Make Selection – When all the options have been carefully researched and a template has been used to develop a product score, it is time to make a final selection.

Most software vendors offer either training seminars, trial use periods, or have limited capability versions of their software available for download on their websites. Rather than purchasing the software package with the highest product score, several top contenders should be investigated more closely. In this way intangible features of the software can be subjectively judged firsthand. Then the package best suiting the goals of the simulation project can be selected for usage.

5.3 Model Construction

Models may be constructed using a simulation language, simulator, or simulation software development environment. The construction stage of a simulation involves writing or constructing computer code to accurately represent the system being modeled. Prior to coding, a flow chart or block diagram should be developed as a means of thinking through the model. In addition to being a design tool, flow charts and block diagrams provide formal documentation illustrating a model's logic.

The process of writing the model's code will be somewhat dependent on the simulator or simulation language chosen. However, by this time in a simulation project, the system has a preliminary design, the model has been thought out and conceptually designed and input data has been analyzed. This reduces the actual construction of a model to an almost clerical function. Some creativity must be used to find an appropriate means of representing an entity or process in terms of the constraints imposed by the software being used, but otherwise, model construction is a relatively routine procedure.

Careful Documentation: One of the best forms of verification is the careful documentation of a simulation model. This includes both documenting the programmed code and all the assumptions that have gone into building the model. The documentation process forces the programmer to think through his/her logic one more time and put this thought process down in writing.

Structured Programming Approach: A structured approach to programming involves breaking the model into logical program modules which can each be individually tested, debugged, and ensured to work properly. It is much easier to debug twenty modules, each consisting of fifty lines of code than it is to debug a single thousand line program. If the program were run as one large unit, locating errors would be difficult and time consuming.

5.4 Verification

In order to ensure a model will be representative of its real world counterpart, a verification process should be followed. This process extends through-out most of a simulation development project and normally calls for program debugging and manually checking calculations used in the model. It ensures the computer implementation of the conceptual model is correct.

A number of measures can ease the verification task. These can be grouped into two broad categories: preventative verification and appraisal verification.

Preventative verification strives to ensure that the simulation has been created in a way that will minimize errors. Two methods of preventative verification include:

Appraisal verification is done to check or appraise the programming after it has been coded. Several appraisal methods follow:

Trace: Most major simulation languages have incorporated a tool called a trace. A trace enables each programming path to be checked by printing out the values of key variables, counters, and statistics after the occurrence of each event in the simulation. This feature enables the programmer to ensure proper statistical updates and logical decision making are taking place correctly.

Animation: A very popular method of checking program veracity is through the use of animation. When a graphic animation is available, it becomes possible for a programmer to visually verify system operation by watching the computer screen. Although verification in this fashion is easy, it won't guarantee that no subtle problems exist. It is best to use animation in conjunction with other methods of verification.

Run under Simplified Conditions: A good method of verification is to run the simulation using simplified conditions. Hand calculations can be performed to prove that the model operates as expected.

Apply Common Sense to Output Reports: An often overlooked method of verification is the use of common sense. If something in an output report looks unusual or out of place, it probably is. The more experience a simulation analyst has, the better he/she becomes at finding errors in this way.

5.5 Bibliography

A.M. Law and W.D. Kelton, Simulation Modeling and Analysis, 3rd Edition, McGraw-Hill Book Company (2000).

R.W. McHaney, Computer Simulation: A Practical Perspective, Academic Press, San Diego (1991).

R.W. McHaney, Simulation, In Miriam Drake (Ed.), Encyclopedia of Library and Information Science, Second Edition, Marcel Dekker, New York, 2643–2655 (2003).

T.H. Naylor and J.M. Finger, "Verification of Simulation Models," Management Science (14:2), B92 B101 (1967).

F. Neelamkavil. Computer Simulation and Modelling, John Wiley and Sons, New York (1987).