Computer Simulation

Mark Isken

Computer simulation involves creating a computer software based model of a system for the purpose of studying system behavior in order to further our understanding of the system. It is commonly used when systems have a significant amount of uncertainty or randomness associated with them and when the relationship between system inputs and outputs is not particularly clear. It is a *descriptive* modeling technique in that it is used to predict the performance of a system for a given set of input parameters and system configuration. It does not provide prescriptive or optimal solutions in the sense of minimizing some objective subject to various constraints. The onus is on the modelers and analysts to generate the scenarios to be simulated and compared based on some set of performance measures. For example, simulation modeling could be used to predict the effect on accounts receivable and staffing utilization for various proposed business process changes involving billing systems and the related information technology and human resources. However, simulation will not tell you the level of resources needed; the decision makers must run the simulation model for different staffing levels and process change scenarios and then compare the results in order make a more informed decision.

Simulation differs from analytical modeling (e.g. regression, queueing, or optimization) in that it is less abstract, or mathematical, and it is easier to conceptualize the link between the model and the real system. Instead of modeling a system with mathematical equations, simulation modeling is more like building a dynamic, electronic scale model of a system. Individual entities making up business transactions are modeled as they move through the system, compete for resources, and eventually exit the system. In order to capture the natural variability of these systems, computer simulation relies on using *pseudo-random numbers* to generate an artificial history of the system in operation. For example, as part of a network server simulation model, both the timing of arrival requests to the server and the specific computing resources requested are determined by sampling from probability distributions that have been selected by the modeler as being representative of the real system. Simulation is essentially a method for conducting highly controlled statistical experiments on a model of a real system. As such, the output of a set of simulation experiments must be analyzed statistically in order to draw inferences regarding the system being studied.

Three broad classes of simulation modeling approaches include:

1. Discrete event simulation
2. Monte-carlo simulation
3. Continuous simulation

Discrete Event Simulation

Systems are modeled as a collection of discrete entities, locations, and resources which interact as part of some business process as time elapses. Entites may, among other things, visit locations, compete for resources, be joined with other entities, be transformed in some way and eventually exit the system. Such events take place at discrete points in time as the simulation progresses. Discrete event modeling is commonly used for many business processes including manufacturing of discrete items, hospitals, call centers, office operations, and computer and communications networks.

Some of the widely used discrete event simulation packages include:

Arena – Systems Modeling Corporation (http://www.sm.com/)
CSIM18 – Mesquite Software, Inc. (http://www.mesquite.com/)
Extend – Imagine That, Inc. (http://www.imaginethatinc.com/)
GPSS/H – Wolverine Software Corporation (http://www.wolverinesoftware.com/)
iGrafx Process – Micrografx, Inc. (http://www.micrografx.com/)
Micro Saint – Micro Analysis & Design (http://www.maad.com/)
ProcessModel – ProcessModel, Inc. (http://www.processmodel.com/)
Promodel/MedModel – PROMODEL Corporation (http://www.promodel.com/)
Silk – ThreadTec, Inc. (http://www.threadtec.com/)
SIMSCRIPT II.5 and SIMPROCESS – CACI Products Company (http://www.caci.com/)
Simul8 – Simul8 Corporation (http://www.SIMUL8.com/)
Taylor ED – F&H Simulations, Inc. (http://www.taylor-ed.com/)
Witness – Lanner Group (http://www.lanner.com/corporate/)

Monte-carlo Simulation

Monte-Carlo simulation is a term used to describe simulation models in which the passage of time is not particularly meaningful. For example, one might build an elaborate financial model using a spreadsheet. If certain quantities in that model such as, say, interest rates or cash flows, were to be replaced by probability distributions, the outputs of the financial model would become random variables. These random variables can be simulated by repeatedly recalculating the spreadsheet, each time resampling from all the input probability distributions used in the model. The popular spreadsheet add-ins, @Risk (http://www.palisade.com/) and Crystal Ball (http://www.decisioneering.com/) facilitate this type of simulation modeling. It has enormous potential given the huge number of static spreadsheet models that analysts of all types have created in many different industries. Its primary impact is in moving analysts away from the common (and dangerous) approach of ignoring uncertainty, using point estimates of highly uncertain quantities, and treating the result of the deterministic calculations as “the answer”.

Continuous Simulation

Systems are conceptualized as stocks and flows and are modeled using differential equations. System dynamics (Sterman, 2000) is one continuous simulation approach that has found wide use in business. Tools such as Powersim (http://www.powersim.com/) and Vensim (http://www.vensim.com/) allow modelers to build systems dynamics models without explicitly using differential equations.

Simulation Applications in Information Systems

Simulation can be used to study information systems related phenomena such as IT enabled business processes (Shwartz and Weber, 1997), web server caching (Watson, Shi, and Chen, 1999), internet pricing (Gupta et. al. 2000), survivability of computer and communications networks (Moitra and Konda, 2000), supply chains (Jain et. al. 2001), software project management, software development and testing (Reddy, 2000), staffing and scheduling of help desk personnel or service technicians (Watson et. al. 1998), along with many others. Computer simulation has long been widely used for capacity planning and performance modeling of information systems and technology, particularly for computer and communications networks (Menasce and Almeida, 1998 and Jain, 1991). A good source for papers and case studies involving simulation is the Proceedings of the Winter Simulation Conference. Full text papers are available at http://www.informs-cs.org/wscpapers.html.

It should be noted that analytical (mathematical) modeling techniques such as queueing analysis, reliability analysis, Markov models and differential/difference equations are also widely used to model some of the same systems or business processes listed above. The choice between using simulation modeling versus analytical models depends on many factors including the complexity of the system being modeled, the nature of the outputs desired, the degree of data availability, and mathematical tractability. While simulation is very flexible and can model extremely complex situations, simple analytical models may offer great insight while needing far less data and be more amenable to embedding in sensitivity analysis or optimization frameworks. On the other hand, with analytical models the mathematics may become quite involved and simulation may be the only viable alternative for modeling and predicting system performance. Analytical and simulation modeling are often combined. Large simulation models may contain submodels which are analytical in nature (e.g. regression models). In the popular systems dynamics (Sterman, 2000) approach, systems are modeled with differential/difference equations which are then simulated and the results displayed visually.

Why and When to Simulate

In Harrell et al. (1999), the following answers are given to the question "why simulate?"

- Captures system interdependencies,
- Accounts for variability in the system,
- Is versatile enough to model any system,
- Shows behavior over time,
- Is less costly, time consuming, and disruptive than experimenting on the actual system,

- Provides information on multiple performance measures,
- Is visually appealing and engages people's interest,
- Provides results that are easy to understand and communicate,
- Runs in compressed, real, or even delayed time,
- Forces attention to detail in a design.

This last point should not be underemphasized. The process of building a simulation model requires detailed exploration, documentation and explanation of the processes involved in the system being studied. The understanding of the current system gained through this exercise can be invaluable. It may even spur system improvement ideas that eliminate the need to actually build the simulation model!

Harrell et al. (1999) also address the related question of "when to simulate?"

- An operational (logical or quantitative) decision is being made,
- The process being analyzed is well defined and repetitive,
- Activities and events exhibit some interdependency and variability,
- The cost impact of the decision is greater than the cost of doing the simulation,
- Cost to experiment on real system is greater than cost to do a simulation.

Components of a Simulation Study

The following has been adapted from two popular simulation textbooks: Law and Kelton (2000), and Banks,Carson and Nelson (1999). Both of these textbooks provide very comprehensive treatments of the field of computer simulation. An outline for a simulation study might involve the following stages:

1. Problem identification
2. Objective Setting and Project Planning
3. Conceptual Model Design
4. Model Life Cycle Design
5. Data Collection
6. Model Development
7. Model Verification
8. Model Validation
9. Experimental Design
10. Production Runs
11. Analysis and Results Reporting

Problem Identification

The first questions to ask when considering building a simulation model are: "What's the problem we're trying to solve? What performance measures are we trying to predict or estimate?" Simulation models should be developed with a clear vision of the problem or class of problems to address. It can be dangerous to try to build a "universal model" of a large system with the goal of being able to address any problems that happen to arise in the future. The scope, level of granularity or detail of the model, data requirements and the model architecture will all be affected by the nature of the problem to be solved. Force yourself to write down the specific questions for which you are hoping to find answers for with your future model. Even go so far as to mock up graphs and summary tables that you plan on creating from the simulation output. This will force you to think about those key factors that absolutely must be in your model and will reinforce a focus on the problem to be solved rather than the model itself.

Objective Setting and Project Planning

Like any analysis and software development project, simulation projects require planning and management. After identifying the primary problems to be addressed, you need to develop concrete objectives for the simulation project itself. What are the deliverables? Is it purely analysis and summary reports or is the model itself to be a final product? What role will these deliverables play in the greater context of the managerial problem being addressed. When doing objective setting you need to consider the general scope of the simulation model. What will be "in the model" and what will not? Scope definition will define in some sense what questions can and cannot be answered by the simulation analysis. Other decisions made at this stage include selecting the simulation software to be used, the project time frame and the members of the project team. In a typical simulation study, the project team might include the simulation modelers/analysts, representatives from relevant operational areas, information systems people and financial analysts. As with problem identification, broad participation by project stakeholders is important at this stage as it acts as the foundation for the entire project.

Conceptual Model Design

Conceptual model design takes place well before actual model software development begins. Model design can be difficult and requires expertise in the craft of modeling. The goal is to abstract the essence of the system being studied so that a reasonably accurate model of it can be created. Important input, output and decision variables should be identified as well as assumptions as to how the system operates. In general, it is advisable to start with the simplest model possible and add complexity as needed to capture necessary features of the system being studied. The level of detail for the model should be identified. For example, a model of a wide area network may

Pictures can be very helpful at this stage. High level graphical representations of the model architecture (see Figure 1 for an example of an outpatient clinic model) can help to define the scope of the model including the key inputs and outputs. Flowcharts are useful for representing detailed processing logic.

Model Life Cycle Design

Is your model for a one time problem or will it be used on an ongoing basis? If ongoing use is envisioned, will it be used regularly or on as needed basis? Who will use the model, an operations analysis or an someone with little or no simulation knowledge? How will the model be maintained and updated? All of these questions are related to the life cycle of the proposed model. It is a very different matter to develop a model that will be used solely by simulation specialists for a one time study than it is to develop a model that can be distributed to and used by managers or other decision makers. The latter requires significant effort to provide end users with an intuitive interface which allows them to interact with the model yet prevents them from inadvertently modifying critical model constructs. Also, as we will discuss shortly, analyzing the output of simulation models requires a certain amount of statistical know-how and this must be automated if you wish to minimize the chances of users drawing invalid conclusions from model outputs.

Data Collection

Data collection is often done in parallel with model development but we list it first to emphasize that you must give some thought early on to how you will go about obtaining the data necessary to populate the model you've envisioned and designed. Most models will have a number of input parameters and distributions for which data is needed. This is called *input modeling* or *input distribution fitting*. Ideally, much of the data is available from various computerized information systems. Unfortunately, it is very often the case that information systems do not capture the detailed process data that is so often needed for computer simulation. In this case, manual data collection is needed and might involve observation and time study, medical record abstracting, or logging by staff members.

While the primary purpose of the data collection effort is for input modeling, one should also collect some data corresponding to system performance measures for which the model will be used to predict. This data can then be used as part of the model validation process. For example, for an appointment call center model you may be able to obtain call durations and volumes from an automatic call distributor (ACD) system. Such a system also should have data related to the amount of time a patient spends on hold waiting to talk to a staff member. After model development is done, the simulation model could be used to predict caller hold times at these known call volumes and durations and the predictions compared to the actual hold times as collected from the ACD.

Model Development

This is the phase in which you actually build the model using either a general purpose programming language, or more likely, a dedicated simulation development package. In order to build robust, easy to use and maintainable models, several practices are suggested. Try to separate model logic from data used by the model as much as possible. Many simulation packages facilitate this by allowing data to be read in from spreadsheets, databases or text files. Separation makes it easier to model many scenarios, minimizes inadvertent errors due to frequent model modifications, and obviates the need to create and manage many versions of the same model which differ only by data parameter values. This stage may also include development of supporting "software machinery" (e.g. using spreadsheets or database management software) for managing model inputs and outputs as well as for creating a custom user interface to the model. Use sound software engineering practices such as commenting your code and using variable naming conventions. Learn to use the debugging tools (including animation) which are available with most simulation languages.

Keep the client involved. Avoid the temptation to “go away and build the world’s greatest model.” Ongoing dialogue with the project team helps to maintain project momentum and contributes to building a valid model by forcing ongoing checks and revisions to assumptions and modeling decisions. While doing model development you should be starting the next two stages of the study, verification and validation.

Model Verification

Does the programmed model operate correctly and as you designed it? Techniques for verification include simulating extreme cases (such as replacing all random elements in the model with single point estimates) in which it is possible to predict or calculate output measures without the simulation model. The current generation of simulation development environments also include tools to facilitate verification including animation, tracing detailed model logic, debuggers, and the ability to write out detailed log files.

Model Validation

Does the model provide a sufficiently accurate approximation to the real system being studied so that meaningful conclusions can be drawn with relatively high confidence? There are a number of ways to assess and improve the validity of a simulation model. However, what finally matters the most is that the output of the simulation model is sufficiently close to the output of the real system it represents. Thus, if the new model can be used to model an existing system, one can collect both input and output data for the existing system. The input data can be used as a basis for simulating the current system and the simulated output can be statistically compared with output data from the actual system. Other methods which contribute to creating a valid model include (see Law and Kelton (2000) for further discussion):

- Documenting modeling assumptions and model logic,
- Structured walk-through,
- Interacting regularly with system experts,
- Collecting high quality data and understanding its origination.

Experimental Design

You may have many input (independent) variables with which you want to experiment and several output measures of interest. This can result in a prohibitively large number of simulation runs. Design of experiments can help reduce the number of experimental combinations needed while still providing sufficient output stats for estimating performance measures of interest. Related issues include the length and number of simulation runs to do in order to make sound statistical inferences. Most simulation texts discuss design of experiments in the context of simulation analysis.

Production Runs

This involves the actual simulation runs for the different points in the experimental design. Some simulation packages make it easy to automate a set of many runs and many different scenarios. A recent trend in simulation packages is the ability to control the simulation environment from spreadsheets or database packages using a programming language such as Visual Basic for Applications. This can be a tremendously powerful technique for automating a very large experiment including the running the models unattended and automated gathering of output statistics.

Analysis and Results Reporting

Simulation models can produce huge amounts of output data. Care must be taken to use appropriate statistical techniques in analyzing this output. See Law and Kelton (2000) for a thorough treatment of this important topic.

A challenge as an analyst is to condense megabytes of output into meaningful summary reports which can be understand by the client. Rarely do the “canned” reports included in most simulation packages fit this need. Fortunately, many simulation packages make it easy to export both raw and summarized data from simulation runs for analysis using spreadsheets and database management systems.

Again, see texts such as Law and Kelton (2000), and Banks, Carson and Nelson (1999) for a thorough description of each of the stages of conducting a simulation analysis.

References

Banks, J., J.S. Carson, and B.L. Nelson (1999), *Discrete-Event System Simulation*, 2nd *Edition*, Prentice Hall, New Jersey.

Gupta, A., Jukic, B.A., Stahl, D.O. and A.B. Whinston (2000), Extracting Consumers' Private Information for Implementing Incentive-Compatible Internet Traffic Pricing, Journal of Management Information Systems, 17,1 , Summer 2000 , pp. 9 - 30

Harrell, C., Ghosh, B.K., and R. Bowden (2000), *Simulation Using ProModel*, McGraw Hill, Boston, MA.

Jain, R. (1991), *The Art of Computer Systems Performance Analysis*, John Wiley and Sons, New York.

Jain, S., Workman, R.W., Ervin E.C., Collins, L.M. (2001), Analyzing the supply chain for a large logistics operation using simulation, *Proceedings of the 2001 Winter Simulation Conference*, 1123-1128.

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

Menasce, D.A. and Almeida, V.A.F. (1998), *Capacity Planning for Web Performance: Metrics, Models and Methods*, Prentice Hall, NJ.

Moitra, S.D. and Konda, S.L. (2000), "A Simulation Model for Managing Survivability of Networked Information Systems", Technical Report, CMU/SEI-2000-TR-020, ESC-TR-020, http://www.cert.org/research/00tr020.pdf.

Reddy, R. (2000), Building the Unbreakable Chain, Intelligent Enterprise, 3,3, http://www.intelligententerprise.com/000209/feat3.shtml.

Schwartz, R.A. and B.W. Weber (1997), Next-Generation Securities Market Systems: An Experimental Investigation of Quote-Driven and Order-Driven Trading, *Journal of Management Information Systems*, 14,2 57 – 80.

Sterman, J.D. (2000), *Business Dynamics: Systems Thinking and Modeling for a Complex World*, McGraw-Hill/Irwin.

Watson, E.F., Chawda, P.P., McCarthy, B., Drevna, M.J. and R.P. Sadowski (1998), A Simulation Metamodel for Response-Time Planning, *Decision Sciences*, 29,1, 217-241.

Watson, E.F., Shi, Y., and Y. Chen (1999), A user-access model-driven approach to proxy cache performance analysis, *Decision Support Systems*, 25, 309-388.

More Resources

See http://ubmail.ubalt.edu/~harsham/simulation/sim.htm for very nice introduction to systems simulation including examples relevant to information systems. See also http://ubmail.ubalt.edu/~harsham/ref/RefSim.htm .

See http://www.informs-cs.org/wscpapers.html for full text articles from all of the Winter Simulation Conferences dating back to 1997.

August 2003 issue of OR/MS Today contains their periodic simulation software survey. See it at http://lionhrtpub.com/ORMS.shtml.

See Prof. John Sterman's web site at http://web.mit.edu/jsterman/www/. Sterman is a leader in the field of system dynamics.

INFORMS College on Simulation - http://www.informs-cs.org/

Michigan Simulation User Group - http://www.m-sug.org/