

THE ABCs OF SIMULATION PRACTICE

FROM “ARRIVALS” TO “Z-VALUES,” AN ALPHABET SOUP OF KEY COMPONENTS FOR APPLYING KNOWLEDGE AND JUDGMENT TO SIMULATION PROBLEMS.

BY JERRY BANKS AND RANDALL R. GIBSON

IT'S POSSIBLE TO understand the theory of simulation but not know how to practice it to solve a problem in the real world. Simulation practice requires training, experience and professional judgment. This is achieved through time, similar to apprenticeship, during which the trainee is involved with numerous projects from problem definition to writing the final technical report under the close supervision of an experienced simulation consultant or practitioner.

During the mentoring period the trainee observes operations at the site, asks numerous questions about the process and learns how the firm has modeled similar situations. No single approach or methodology works for every project. A successful simulation practitioner relies on experience to construct approaches to each project and challenge.

Judgment is acquired through practice, observing what works, what doesn't, and why. It is perhaps the least understood skill to successful simulation practice, but the most important.

Successful simulation practitioners draw on experience and judgment in a variety of topics. For example, there is the determination of the real problem that needs to be solved, rather than the problem indicated by the symptoms. Another determination is when enough verification and validation has been conducted to accept the simulation model as a substitute for reality. Yet another determination is what data is needed for a study, and what can be ignored. These judgment calls are numerous and continue throughout the problem-solving process.

In the following, we present what we consider to be the “ABCs of Simulation Practice.” Letter by letter, we summarize 26 of what we consider to be important components for applying knowledge and judgment to simulation problems. We don't claim that these are “the” 26 components. Other simulation analysts could have another set, or they could have multiple components for one letter of the alphabet, and none for another letter of the alphabet.

Our comments in this article are not necessarily what you will find in a



simulation textbook. What we are presenting here relates more to the use of simulation in solving real-world problems. Both textbook understanding and practical knowledge are important to success in using simulation. If your desire is for “textbook” information, see Banks, et al. [2005] or Law [2007].

A is for arrivals. Arrivals are the driver of discrete-event simulation models, a very appropriate beginning for this discussion. Since arrivals are such an important input to a simulation, it’s crucial that they are correct. We introduce errors in modeling when we fail to imitate reality in the

input data. If the arrivals vary over time, we need to capture that phenomenon.

B is for baseline. If we are going to redesign a system, first we simulate the old system, i.e., the “as is” or baseline configuration. When we are satisfied that the baseline configuration has been simulated properly, we revise it to simulate the redesigned “to be” system. A simulation of the baseline configuration will help us greatly to understand the way the system operates. In some cases, as the baseline system is simulated, there is sufficient added understanding of the existing system such that the system can be greatly improved without further modeling.

C is for confidence interval. Confidence interval is a code word for conducting statistical output analysis. (We assume here that you have random inputs, which leads to random outputs.) At the most basic level, we are trying to predict the performance of a single design or compare the performance of two or more alternate designs. Next, we make a distinction between terminating simulations and

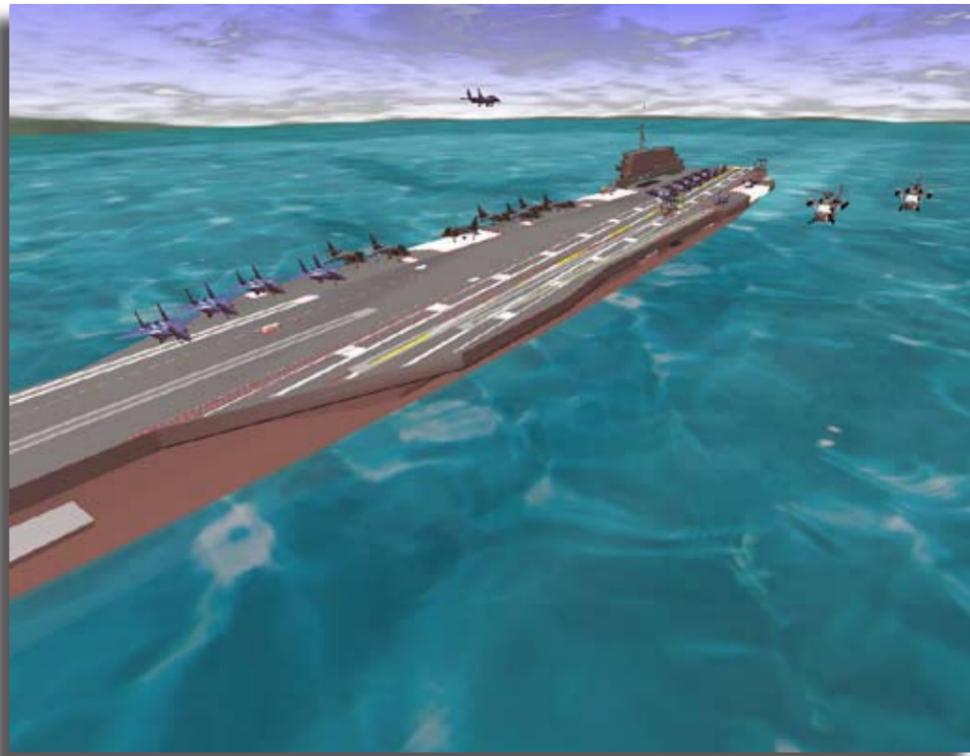
steady-state simulations. Steady state simulations are more difficult to analyze because there are initial conditions and the decision of an appropriate run length is needed. Consider a terminating simulation that runs from time zero to some ending time. We conduct multiple replications of this model, each of them with a possibly different result since there is randomness in the input. Using statistical methods or statistical software we find the desired confidence interval, usually 95% or 99%.

D is for data. Data collection is one of the biggest tasks in solving a real problem. (In addition to inter-arrival time data there are service times, time to fail, time to repair, fraction of entities going in one of several directions, travel speeds and many others). It’s also one of the most important steps in the simulation process. Remember the term “GIGO.” It applies to simulation input data. That is, garbage-in, garbage-out. Even when the model structure is valid, if the input data has been collected inaccurately, or inappropriately analyzed the output can be misleading or simply wrong leading to decisions that are costly or worse.

E is for experimentation. We emphasize that the simulation analyst go

to the extremes in experimentation. Every queue should be examined to ensure that it is not growing without bound and that it is not always empty. If it is growing without bound the simulation is unstable. An empty queue could mean that the model structure is in error, or it could mean that the input data is in error. We also suggest constructing experiments that will bring insight to the simulation analyst. The model must be stressed. That is, let the input values increase until the model becomes unstable to make sure that the structure of the model is appropriate.

F is for fidelity. Fidelity is how closely the model can mimic the behavior and operation of the real system (think of the fidelity of an audio system – how faithfully it can reproduce the original sounds recorded). The fidelity that is required of a model is a function of what questions we are trying to answer using the model, and is related to the level of detail in the model and the resolution (accuracy) of the input data. Increases in model fidelity come at a steep price in terms of the effort needed to build and test the model. The analyst needs to exercise judgment when deciding what fidelity is required and what resolution in the data is required.



G is for graphics. It is said that “a picture is worth 1,000 words.” But, we also say that “one good analysis is worth 1,000 pictures.” We warn you not to be swayed by graphics that are fancier than need be for the situation. The purpose of graphics, in our opinion, is twofold. First, graphics helps the simulation analyst to see modeling errors. Second, graphics can be a selling tool for the system that is being proposed. Rather than making your decision based on the graphics provided, make it more on the power of the simulation to solve the intended problem.

H is for histograms. Simple data graphics can be very helpful in understanding the system that is being simulated. A histogram is a simple graphic. It has bins on the horizontal axis and frequencies or probabilities on the vertical axis. These can be drawn readily using Excel. In addition to histograms, simple graphics include scatter plots, cause and effect diagrams, Pareto charts, and bar-and-whisker diagrams. These are readily understood and we recommend them highly to explain the input data or output data.

I is for initial conditions. Consider a single run of a simulation model whose purpose is to estimate the steady-state behavior of the system. The system starts idle and empty. It will take a while for the system to reach its steady state where the queues are filled approximately to their long-term average values. If we begin our statistical data collection at time zero, there will be initialization bias in the performance measure. There are several options here. The first option is to initialize the system with what we believe would be a realistic number of entities being served and in queues. We would also have to place some machines in a failure mode. The second option is to divide the output data into two parts: 1. the warm-up part beginning at time zero and ending some time later, say t^* ; and 2. the steady state portion beginning at time t^* and ending at the endpoint of the simulation. This second method is used very often in the practice of simulation.

J is for Java. Over the years, many simulation models have been written in a programming language. First it was FORTRAN, then C, then C++ and now Java. However, general programming languages do not provide simulation support functions including event

scheduling, the system clock and the time advance algorithm. Simulation products have also been written in these programming languages, which include these support functions. For the models written in Java, there is the advantage that a stand-alone simulation model can be delivered to the client for use on any Windows computer.

K is for knowledge. You need to know the system that you are simulating. You need to ask a lot of questions. Remember, no two people will have the same perspective. Collect lots of information; for example, some CAD drawings can be used as the basis for the graphic animation of a system being simulated. Some of the resources on the CAD drawing can be directly converted to resources in the simulation model, saving lots of time, enhancing accuracy and making for a better animation. After gathering this knowledge, prepare the assumptions that you will use in the simulation model, and have the client agree that the assumptions are accurate.



L is for linearity. We suggest that you look for linearity, but find the point where it disappears. An example is an automated guided vehicle (AGV) system. Perhaps, two AGVs are better than one. But three AGVs might not be better than two as the system gets clogged with AGVs waiting for an assignment. Similarly, faster conveyors can reach a point where merges become impossible. So, understand that linearity often has limits.

M is for manufacturing and material handling. A great amount of simulation is devoted to these two areas, for good reason. As much as 80 percent of the time between raw material and finished product states, goods are in transit or waiting for transit, i.e., in

SUBSCRIBE TO ANALYTICS

It's fast, it's easy and it's FREE!
Just visit: <http://analytics.informs.org/>

the material handling system. This represents a very large investment of capital that could have been invested in some alternative income-producing asset. Many simulation projects are intended to solve problems in manufacturing and material handling systems.

N is for nature of the underlying process to be modeled. It is critical for the analyst to have sufficient understanding of a process in order to correctly model the data values it will take on in the simulation. For example, if the Poisson process describes the arrivals, then the time between arrivals is exponentially distributed. On the other hand, when no data is available, understanding the underlying process allows for some initial guesses for input distributions.

O is for object orientation. Most of the newer simulation software products are built using object-oriented programming systems or are themselves object-based systems, in which users can modify pre-defined standard objects to construct custom objects in a hierarchical model of their particular application. The software objects mimic the actions or

behavior of their real-world counterparts (e.g., a machine), and interact with each other using “messages.” Object-based systems are a much more natural way to represent real-world systems in a model, but require a different approach to the design and development than traditional procedural based languages or simulation systems. Object-based simulation systems appear to be the direction for the future and will require the analyst to be familiar with this new approach to designing and building models.

P is for programming. Among the most important elements that contribute to the validity and usefulness of a model is the level of detail and complexity included to represent the business decision rules, control algorithms or other control logic for the specific system being analyzed. Having the flexibility to get away from the often limiting built-in point-and-click and drag-and-drop rule set to write custom logic that closer approximates the real system logic is what gives power to model the real system. Programming is the process of designing, building and testing custom logic – regardless of the language or method of implementation.

DISCOVERIES +
BREAKTHROUGHS
INSIDE
SCIENCE
features informs



Got an O.R. Success Story for the Nightly News?

INFORMS is now a partner in Discoveries and Breakthroughs Inside Science (DBIS), a science media program founded, in part, with National Science Foundation funding. The new alliance permits INFORMS to suggest story ideas for production and distribution as television news items.

Do you have a story idea that you'd like to share with DBIS? Then read on!

About DBIS

DBIS provides twelve 90-second reports (in either English or Spanish) to subscribing local TV newscasts all over the United States. Accurate, scientifically reviewed and reliable, DBIS brings the latest science, engineering, math and technology news to a general audience that is underserved with quality science reporting.

Supported by a coalition of scientific and engineering professional societies and a generous contribution from the National Science Foundation, DBIS promotes awareness of and appreciation for the role science, technology, engineering, and mathematics professionals play in our daily lives.

Submit Your Story Ideas

DBIS has 23 Partnering societies. Now that INFORMS has joined, INFORMS members have a unique opportunity to bring attention to our field, especially to research that has strong practical and visual qualities.



DBIS Story Criteria

- Are the researchers located in the contiguous United States?
- Does the story idea affect people across the country (not regionally limited)?
- Is the idea of high interest to the general public and not just researchers in that field?
- Does the story have elements that we can see or show?
- Is there an outside expert available to review the research?
- Is the medical research in at least Phase II of human clinical trials?
- Will the technology be used or will there be a product available within the next year?
- Is the researcher noteworthy both professionally and personally based on their career?
- Has the research already received television news coverage?

If you can say "yes" to at least three of these, contact INFORMS Communications Director Barry List at barry.list@informs.org, 1-800-4INFORMS, or 443-757-3560 to discuss your story idea. Broadcast your important research today!

Q is for queueing. The measurement of the number or time in queues (e.g., waiting lines) is the major interest in discrete-system simulation. In some simple cases we can approximate these measures. But, in most real cases, the mathematical models are not sufficiently descriptive, and we must resort to simulation to approximate the queue measures. An example is a queueing system in which the number of servers varies by the number of customers waiting in the queue and the resources break down randomly and also require preventive maintenance.

R is for run length. When we run a simulation, we select the time span for the analysis period – e.g., a 24-hour period for a manufacturing process, maybe a seven-day period for a mail sorting facility or 360 days for a supply chain network. It's up to the analyst to set this time span to represent the “period of interest” that will support whatever analysis needs to be done. Deciding on this interval depends on a variety of factors including the following: (a) understanding the underlying nature of the system such as natural repeating cycles, (b)

analysis objectives and how those translate into the period of interest - e.g., the busiest day in the week for a mail facility), (c) what, if any, warm-up period is required and how to estimate this, (d) unscheduled events that may occur during this time span and which need to be accounted for, and (e) the number of observations that the model will collect from key events that may occur during this time span.

S is for sensitivity testing. Sensitivity testing is a method for determining the relative importance and impact on key model output variables of input variables or distributions of processes for which we have limited or no data from which to construct a distribution to represent the process. This method involves making rough estimates of the range over which the variable may take on values, and some notion of the underlying process that will allow an initial selection of a distribution, even if only uniform or triangular. We then run several replications of a series of experiments, each using different values from within the estimated range for the input process in question. If examination of the key output variables shows little change in response to the range of input values for these processes, we may conclude that this particular model is not sensitive to this process, eliminating the

need to obtain more accurate estimates of the input values or underlying distribution. If, on the other hand, the output values show considerable swings, we conclude that the model is very sensitive to this process, and we must then develop more representative input data.

T is for teamwork. A successful simulation project is the result of teamwork including planning, collaboration and communication between the various project participants. Getting a project done on schedule doesn't just happen. The project must be planned and executed properly. Good communication is required for the analyst to understand details of the system to be simulated, and for the client to develop realistic expectations of the simulation process. Both parties collaborate on deciding what is included, what is not, at what level of detail and when the analysis is completed. The analyst must take the lead in establishing the requirements and methodology for this teamwork process, and to actively guide or even drive it to completion.

U is for unscheduled events. Unscheduled events include breakdowns or other unplanned interruptions. In general these can be very complex and difficult to correctly model. The typical approach is

to simply collect process data, and then generate these events in the model using an exponential distribution whose mean is based on the observed data. In practice, however, we find that this can have unintended consequences due to the potentially large values that can be generated by this distribution – often much longer than the period of interest to simulate. A more practical way to do this includes separating the unscheduled events into two classes. The first class includes events that typically occur one or more times within the timeframe of interest to simulate (e.g., one week for a manufacturing process), such as conveyor jams or robot faults. The second class includes rare events, such as major machine breakdowns, which occur much less frequently on average than the simulated timeframe. First, run the model with neither class of unscheduled events turned on, to understand the basic behavior or capacity, etc. of the system without interruptions. Second, turn on the first or “typical” unscheduled events and re-run the model to determine the impact of these events, which must be included as part of any analysis. Finally, instead of allowing a distribution to generate random times for the rare unscheduled events – which may not occur for a long time, perhaps skewing the analysis, we recommend forcing one

SUBSCRIBE TO ANALYTICS

It's fast, it's easy and it's FREE!
Just visit: <http://analytics.informs.org/>

or more of these events in each model experiment (at a random time within the normal timeframe of interest), and then examining the impact on the system and how long is required to recover from the event. We find this technique more expedient, quicker and certainly easier to replicate.

V is for verification and validation. V/V are essential to successful simulation modeling. Verification asks the question, “Did we model it correctly?” Validation asks the question, “Do we have the correct model?” Verification techniques were borrowed from software engineering. Validation techniques that are used, however, are more unique to simulation. We don’t ever prove that a model is valid. “Is the output reasonable?” is the first question that should be asked. Amazingly, a simulation analyst can get lost in the detail and never look at the output with this question in mind. V/V should be conducted throughout the process, not just at the end. Resources, say 20% to 25% of the project budget, should be planned to conduct the appropriate amount of V/V that is requisite to releasing a model as a representation of the real system.

W is for WSC. The Winter Simulation Conference (WSC) is held in early

December every year. It is the premier conference for discrete-event simulation with some 700 to 900 attendees, depending on location and the economy. For anyone interested in discrete-event simulation, there is always something available with as many as 18 concurrent sessions underway. These sessions range from introductory to advanced and from practice to theory. There are also exhibits where the vendors present their wares. For more info, visit www.wintersim.org/.

X is for the X factor – the unknown. In every project, no matter how carefully planned, there is always something that causes a problem that is a surprise. It can be minor, such as an overlooked step in the process being simulated, or major such as not being able to understand or explain why the model is producing completely counter-intuitive results. The experienced analyst is able to step back from the project – from the model details, from the project assumptions, from the confident way the client stated the key decision control logic – and look for the X Factor that would account for the surprise. There is both art and science to this ability – insight, followed by a logical and methodical approach to testing, eliminating and finally identifying the underlying cause.

Y is for YANSL. YANSL (Yet another network simulation language) is an acronym coined by Joines and Roberts [1997], but we use it here to describe the proliferation of simulation software. For example, nearly 20 simulation software vendors, some with multiple packages for sale, were represented at WSC’08 (www.wintersim.org/ExhGuide08.pdf). For the person new to simulation, this provides a very large choice set.

Z is for Z-values ... and other statistical values as a reminder of the output analysis capability that is available for simulation. These output analysis capabilities are built into many simulation software packages. They perform a variety of functions including statistical analysis, optimization, support for experimentation and output data management. The problem is that many simulation analysts don’t use the capability that is available. Perhaps, they don’t understand what is happening, so they would be hard pressed to explain it to their manager.

CONCLUSION

We consider the 26 items that we discussed to be important, but not necessarily unique. Although many of

these items should be part of every simulation professionals’ experience base, others may have items that are unique to their practice. What is important is to note that to be successful in the practice of simulation, a knowledge base built on practical experience and professional judgment, such as is represented by these 26 items, is critical in the real world. ■

Jerry Banks (atljerry@earthlink.net) holds the title “academic leader” at Tecnológico de Monterrey in Monterrey, México. Retired from the faculty of the School of Industrial and Systems Engineering at Georgia Tech, Banks was formerly a senior simulation technology advisor at Brooks Automation (now Applied Materials).

Randall Gibson (rrgibson@transystems.com) is a principal and senior vice president in the Management and Supply Chain Consulting group at TranSystems Corporation, and manages the Simulation and Analytical Modeling Practice Area. He was previously founder and president of Automation Associates, which was acquired by TranSystems in 2005. He has more than 20 years experience in simulation modeling and consulting.

REFERENCES

1. Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol, 2005, “Discrete-event System Simulation” (fourth edition), Upper Saddle River, N. J.: Prentice-Hall.
2. Joines, J.A., and S.D. Roberts, “An introduction to object-oriented simulation in C++,” Proceedings of the 1997 Winter Simulation Conference, eds. S Andradóttir, K.J. Healy, D.H. Withers, and B.L. Nelson.
3. Law, A. M., 2007, “Simulation Modeling and Analysis” (fourth edition), McGraw-Hill, New York.
4. L’Ecuyer, P., 1998, “Random number generation,” Chapter 4 in Banks, J., “Handbook of Simulation,” Wiley, New York.
5. Lehmer, D.H., 1951, “Proceedings of the Second Symposium on Large-Scale Digital Computing Machinery,” Harvard University Press, Cambridge, Mass.