Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

SIMULATION ATTACKS MANUFACTURING CHALLENGES

Edward J. Williams

PMC
15726 Michigan Avenue
Dearborn, MI 48126 USA

ABSTRACT

All during the past half-century, the environment of computing applications has evolved from large, comparatively slow mainframes with storage small and expensive by today's standards to desktops, laptops, cloud computing, fast computation, graphical capabilities, and capacious flash drives carried in pocket or purse. All this time, discrete-event process simulation has steadily grown in power, ease of application, availability of expertise, and breadth of applications to business challenges in manufacturing, supply chain operations, health care, call centers, retailing, transport networks, and more. Manufacturing applications were among the first, and are now among the most frequent and most beneficial, applications of simulation. In this paper, the road, from newcomer to simulation in manufacturing to contented beneficiary of its regular and routine use, is mapped and signposted.

1 INTRODUCTION

As the world becomes conceptually smaller and more tightly integrated in the economic sense, the challenges of designing, staffing, equipping, and operating a manufacturing process or plant intensify. These challenges include, but are surely not limited to, process design and configuration, selection of personnel (staffing levels and skill levels), selection of machines, sizing and placement of buffers, production scheduling, capacity planning, implementation of material handling, and choices for ongoing process revision and improvement (Jacobs et al. 2011). During its fifty-year history of application to manufacturing operations, simulation has successfully addressed all of these and more (Rohrer 1998). Additionally, simulation correctly used is a powerful force for organizational learning (Stansfield, Massey, and Jamison 2014).

Therefore, let us next examine typical reasons and motivations frequently set forth to initiate a manufacturing-context simulation project:

- We already know the system design we are determined to use, but upper management won't let us spend the money until we do a simulation providing good news about that design. *(Definitely contraindicated!*)
- We have a design (or several) sketched "on a cocktail napkin," and expect simulation to give insight as to its (their) potential capability and indicate points amenable to improvement.
- Our system is already operational, but not satisfactorily; several improvements have been suggested – indeed, have been hotly debated. We need to investigate their merits, both individually and in combination.
- Our system is already operational, and we need to have contingency plans in place for reasons such as increased product demand, increased economic pressures, wider variety of product mix, and/or other plausible changes.

Observe the zero prefacing the first motivation. Beginning a simulation project with this motivation is setting foot on the road to ruin – the simulation results will inevitably be irretrievably contaminated by

978-1-4799-7486-3/14/$31.00 ©2014 IEEE

bias. The next motivation is the one with the greatest potential return on investment (ROI) relative to the cost of the simulation. Many examples exist of a 10:1 ROI, occasionally reaching 100:1 ROI (Murty et al. 2005). In this situation, estimating all needed input data required for the simulation will surely be a challenge – after all, the system does not yet exist! The power of sensitivity analysis (to be explained below) is then extremely valuable. In the last two situations, input data for the existing system, to be modeled as a baseline, will be more readily (not necessarily easily) available. Very possibly, suggested improvement A will be of little value, suggested improvement B will be of little value, yet implementation of both A and B will be of great value. Statistical analysis of the output can expose such valuable insights. "Unsatisfactory operation" may refer to any or all of low throughput, low utilization of expensive resources, excessive in-process inventory, or long makespan (likely including long waits in queues). As examples of such applications, Habanicht and Mönch (2007) achieved improvements to long makespan in a wafer fabrication facility; Khalili and Zahedi (2013) used simulation to prepare a mattress production line for anticipated demand increases over a five-year planning horizon.

2 THE FOUNDATION – WHERE ARE WE GOING?

First, and vitally, when a simulation project is to be started, the following questions must be asked and answered:

- What is to be modeled?
- What questions shall the model and output analysis of it answer, and what decisions will the results guide?
- When are results needed?
- Who will do the work, if it is to be done at all?

Let us explore likely answers to these questions. For question #1, and especially for a first or early foray into simulation usage (which management may be approaching charily), the preferred answer is a small one. Extensive experience strongly suggests that an answer such as "the milling department" or "the XYZ line" augurs much better for eventual success than an answer such as "the whole factory," or, worse, "the whole factory plus inbound and outbound shipments." For question #2, example answers (note that these answers are themselves questions) might be:

1. Of the three proposed alternatives for production line expansion, which one will produce the greatest throughput per hour?
2. Will a specific proposal for line design be able to produce at least 55 jobs per hour?
3. What level of staffing of machine repairpersons will ensure that the total value of in-line inventory will not exceed $40,000 at any time during one month of scheduled production?
4. Will the utilization of a particular critical and expensive piece of equipment be between 80% and 90%?
5. Which of several proposed designs, if any, will ensure that no part waits more than 8 minutes to enter the brazing oven?

Raising and documenting these questions accomplishes several vital tasks. First, these questions will in due course provide an unequivocal basis for answering the final question "Has the simulation project successfully met its objectives?" Second, the questions guide decisions concerning the answer to question #4 above, the level of detail to be incorporated into the model (this level should be as low as possible consistent with answering the chosen questions), guide data collection efforts, and help guide the choice of simulation software. Third, for question #3, typical example answers are:

1. The simulation modeling and analysis must be complete by August 24 for review. Management will make an irrevocable decision on system design on August 31. Results available later than August 24 will be useless and ignored.
2. The sooner results are available, the sooner the company can start earning greater profits via an improved system. It would be nice if results are available by June 27, in time for discussion at the quarterly management review meeting.

In the first case, the project plan will almost surely require modification. Possible modifications include canceling the project (yes!), reducing its scope, adding headcount to the project at its inception (quite dangerous, c.f. "we need the baby in three months, not nine, so three women will be assigned to produce it"), or adding headcount to the project after it is underway (even more dangerous). The last alternative is likely to crash into the figurative iceberg so aptly described by Brooks: "Adding headcount to a late project makes it later" (Brooks 1995). The second case is much more amenable to favoring quality over speed. Fourth, relative to the last question, reasonable alternatives are:

1. Doing the simulation modeling and analysis in-house.
2. Contracting with a service vendor to do this and all future simulation projects.
3. Contracting with a service vendor to do this project, instructing us meanwhile so future projects can be done internally (perhaps with external guidance from specialists).

Now, if the project is to proceed, it's time for data collection.

3 DATA COLLECTION AND ANALYSIS

Data collection is notoriously the hardest and most tedious, time-consuming, and pitfall-prone phase of a simulation project (Sadowski and Grabau 2000). First, consider the wide variety of data typically needed for a manufacturing simulation:

- Cycle times of automatic or semi-automatic machines; process time on manually operated machines.
- Changeover times of machines, whether occasioned by product change ("next one is green, not red"), cycle count (after making 55th part, sharpen the drill bit), working time ("after polishing for 210 minutes, replenish the abrasive"), or elapsed time ("it's been 3 hours since we last recharged the battery").
- Frequency and durations of downtimes; whether downtimes are predicted by operating time, total elapsed time, or number of operations undertaken; whether a downtime ruins or damages a work item in process.
- Travel time, load time, unload time, routes, and availability of material-handling equipment (conveyors, tug-trains, AGVs, forklifts….); whether travel time differs for loaded versus unloaded vehicles; accelerations and decelerations may also be significant and important.
- Frequency of defective product; whether the defective product is scrapped or repaired.
- Operating schedule – number of shifts run, their durations.
- Workers – their schedule, number and type of workers available (operators, repair persons, material-handling workers…), duties, travel time between duties, absenteeism statistics.
- Buffer locations and capacities.
- Availability and frequency of delivery of raw materials.

The author has yet to undertake a manufacturing-simulation project in which the client added nothing to this generic list.

Next, be careful of misunderstandings, such as:

- The client spokesperson said "Cycle time of this machine is 6 minutes." Actually, the operator is needed for 45 seconds to load the machine, which then runs automatically for 4½ minutes, then the operator is needed for 45 seconds to unload the machine; during the 4½ minutes, the operator can travel to/from and perform other tasks. Indeed, the term "cycle time" has no one standard definition.
- The person collecting data reported "The workers work 8am-5pm, with fifteen-minute breaks starting at 10am and 2:30pm and a half-hour lunch at noon." Actually correct, the workers spend 10 minutes (8:00am-8:10am) donning protective clothing and equipment, which they take off from 4:50pm to 5pm.
- The person collecting data reported "The drill press was down for a whole hour, from 9:20am to 10:20am." Actually, the drill press was in working order, but idle, during that time – a problem upstream prevented any work from reaching it.
- The person assigned to collect data during the 4pm-midnight afternoon shift reported the milling machine suffered a 20-minute downtime beginning at 11:40pm. The person assigned to collect data during the midnight-8am night shift reported the milling machine suffered a 20-minute downtime ending at 12:20am. Actually, the milling machine suffered one downtime of 40 minutes' duration.

Forewarned by these examples (all from experience), the reader and practitioner will be alert to others. Further, downtime data are particularly difficult to gather (Williams 1994). Too often, production personnel are reluctant to report downtimes, perhaps fearing that such reports would cast aspersions on the rigor with which maintenance policies and procedures are followed. As another example, a 30-minute downtime might need to be subdivided as (a) the machine was down for 5 minutes before the malfunction was noticed and reported, (b) it took the repair expert 10 minutes to gather needed tools and travel to the machine, and (c) it then took her 15 minutes to effect the repair. Neglecting (a) overestimates the demands on the repairperson.

Next, the input data must be analyzed for best inclusion in the model. For ease of checking and updating the data, practitioners routinely and strongly urge that constant values be kept in spreadsheets (e.g., Microsoft Excel®) and imported into the model (all modern simulation software enables this task), not hard-coded in the model. When data is thus imported into the model, it can be updated without the necessity of updating the model itself. Eliminating this task eliminates the errors introduced by the overconfidence of "I don't know this simulation software very well, but it can't be that hard to open the model and just change a cycle time."

Furthermore, the modeler or analyst must decide whether to use the data directly (i.e., sample from an empirical distribution formed by the data points collected) or fit a closed-form distribution (e.g., exponential, gamma, Erlang, Weibull…) to the data (using readily available software) and sample from this distribution. The latter approach has two significant advantages: (a) it realistically permits sampling values in the simulation which are outside the range of actual data points collected, and (b) it eases the drawing of conclusions concerning the model and its results, since formulas are readily available for common closed-form distributions. However, realizing these advantages is contingent upon finding a closed-form distribution which fits the empirical data well – and that may be impossible. For example, it will be impossible if the empirical distribution is conspicuously bimodal (or multimodal). In that case, re-examine the data. For example, the data set, seemingly "cycle times of the lathe," may really be two data sets: "cycle time of the lathe on x-type parts" and "cycle time of the lathe on y-type parts." In such a case, subdivide the data set and re-analyze each subset. Valuable further detail on distribution-fitting analyses is available in Cheng (1993) and in chapter 6 of Kelton, Smith, and Sturrock (2013). For example, the assessment of how well or poorly the proposed closed-form distribution fits the empirical data may be based upon any or all of the chi-square (also “chi squared”), Anderson-Darling, Cramér-von Mises, or Kolmogorov-Smirnov statistical tests.

Furthermore, looking ahead to the next step, data should be used in the model-under-construction as it is collected. The sooner the data actually enters a model (even one in early stages of development), the sooner significant errors in the data, or misunderstandings involving its collection, will come to light.

4 MODEL BUILDING, VERIFICATION, AND VALIDATION

The task of building the simulation model now waxes large – indeed, in actual practice, data collection and the building of the model should be, and are, undertaken largely concurrently. The choice of software to build the model may be clear if previous simulation projects have been done using that software; here, let us assume that it is not the case (first foray into simulation). Then various considerations might direct the choice of software:

- Use package *x* because its salesperson gave us the flashiest demonstration and made the rosiest promises. (*Definitely contraindicated!*)
- Use package *x* because one or several of our employees have received instruction in its use (perhaps in a university course).
- Use package *x* because our analysts attended a conference where competing packages were exhibited, and those analysts undertook a detailed comparative examination of competing packages relative to our modeling needs.
- Use package *x* because a consultant whom we trust, and who demonstrably has no vested interest in recommending *x*, and who can clearly articulate substantive reasons for choosing *x*, recommends it.
- Use package *x* because of assurance that support (including both software documentation and vendor support) will be timely and of high quality.

The analyst choosing the software must ensure that it accommodates any modeling needs specific to the system to be modeled. Examples of such specific needs might be:

- Ability to model shifts of work, perhaps including situations where parts of the facility run one shift and other parts run two, very likely including situations involving coffee breaks and/or meal breaks.
- Ability to model changeover times, perhaps including situations where more than one cause of changeover (as discussed in data collection) exist.
- Ability to model downtimes whose occurrence is based on any or all of elapsed working time, elapsed total time, or number of cycles executed.
- Ability to model repair operations whose undertaking may be contingent on the availability of a repair person with a specialized skill and/or the availability of specific repair equipment.
- Ability to model bridge cranes, perhaps including multiple cranes and “bump-away” priorities in one bay.
- Ability to model conveyors, accumulating and/or non-accumulating, perhaps including configurations in which the conveyors have curves in which travel speed is reduced.
- Ability to model material-handling operations including equipment such as tug trains, forklifts, high-lows, and/or automatic guided vehicles.
- Ability to model situations in which several parts are joined together permanently in a subassembly.
- Ability to model situations in which expected remaining cycle time suddenly changes because a piece of equipment suddenly becomes (un)available; for example, two polishers working together

need ½ hour more to complete a job, one breaks down, and the estimated remaining cycle time suddenly becomes 1 hour.

- Ability to model situations in which parts are inspected, with some being judged "good" (ready for shipment or use in an assembly), some being judged "needing rework," after which they may become "good," and some being judged "scrap" to be rejected.
- Ability to model situations in which several parts are joined temporarily; for example, to travel together on a pallet.
- Ability to model situations in which several parts are joined permanently, for shipment or for further assembly work.
- Ability to model situations in which raw material or parts are disassembled.
- Ability to interface conveniently with relational database software (e.g., Microsoft Access®).

The task of verification should be concurrent with the task of building the model. Verification, conceptually equivalent with "debugging" in the context of computer software coding, seeks to find and extirpate all errors ("bugs") in the model by testing the model. As clearly stated (Myers 1979), a successful test is one that exposes an error. The analyst should not build the entire model and then begin verification – errors in the model are then difficult to expose and isolate for correction. Rather, the analyst should build the model piecemeal, pausing to verify each component (e.g., another segment of the production line) before building the next component. Verification methods include stepwise examination of the animation (are entities [items] in the model going where they should?) and code or model walkthroughs (the model-builder explains the construction and operation of the model to a willing listener, often becoming aware of an error in doing so).

Validation is fundamentally distinct from verification. Whereas verification answers the question "Is the model built right?", validation answers the question "Did we build the right model?" The right model is one that accurately mirrors the real or proposed system in all ways important to the client, and does so as simply as possible. Therefore, validation requires the participation of the client more than verification does. Powerful methods of validation include (Sargent 1992):

- Comparing model output with observed values of the current system – if the current system exists. Here, a Turing test can be especially valuable. Put similarly formatted reports of model output and of actual system output (e.g., utilizations, throughput, queue lengths and residence times) side-by-side and ask the client "Which is which?" If the client is uncertain, congratulations! The model has passed the Turing test. If the client confidently and correctly says, for example, "The one on the left is the model output," the model fails the Turing test. Ask "How could you tell?" An example answer might be "It has a shorter queue upstream from the milling machine than we've ever seen in practice." Such an explanation provides valuable information for correcting the model and adding to its realism.
- Temporarily replacing all randomness in the model with constants and checking results against spreadsheet computations (also useful in verification).
- Allowing only one entity [item] into the model and examining the output results (also useful in verification).
- Undertaking directional tests: e.g., increasing arrival rates should increase queue lengths, queue residence times, and machine utilizations.
- Checking for face validity: e.g., a chronically long queue directly upstream from a machine with low utilization is suspicious and merits close examination.

The ultimate goal of verification and validation is model credibility. A credible model is one the client trusts to guide managerial decision-making..

5 MODEL EXECUTION AND OUTPUT ANALYSIS

After verification and validation are complete, and the model has achieved credibility in the opinion of the client, it must be executed to evaluate and assess the merits of the system design(s) under investigation. Key questions to ask and answer at this stage of the simulation project are:

1. How much warm-up time is appropriate?
2. How long should the replications be?
3. How many replications should be run?

Warm-up time refers to the simulated time during which the model runs to achieve typical system conditions, as opposed to the time-zero "empty and idle" default condition of the model. To select the warm-up time, the analyst must first decide whether the simulation is "terminating" or "steady-state." A terminating simulation models a system which itself begins "empty and idle," such as a bank. A steady-state simulation models a system which does not periodically empty and shut down, such as a hospital emergency room or a telephone exchange. Most manufacturing systems are steady-state – even if operations pause over the weekend, for example, work very probably resumes Monday morning where it left off Friday afternoon. Whereas terminating systems need and should have zero warm-up time, a model of a manufacturing system must be run for sufficient warm-up time to reach typical long-term conditions before the simulation software is instructed to begin gathering output statistics and performance metrics. Statistical tests are available to help the analyst choose the appropriate warm-up time (Goldsman and Tokol 2000).

The length of a replication (i.e., the simulated time it represents) is likewise a delicate statistical question. The longer each replication is, the more confidence both the analyst and the client can have that the replication will accurately capture representative reality in the system being modeled. One useful rule of thumb is that even the rarest of events (for example, a conveyor breakdown) should have a chance to happen "half a dozen" times during the replication. The analyst does well to remember that the rarest events may be interactions. For example, if each of two particular machines fails occasionally and independently, both machines may be simultaneously "down" very occasionally – yet information on system performance during that situation may be extremely important to have. As an additional convenience, the length of a replication should be an integer multiple of a canonical work period. For example, suppose performance metrics on the actual system are (or will be) gathered on a basis of 24-hour intervals. If the foregoing considerations guide the analyst to a replication length of 450 hours, the replication length might well be increased to 480 hours, representing twenty days.

From a statistical viewpoint, each replication represents another experimental data point – "another throw of the dice" (using different random numbers generated by the simulation software). Therefore, successive replications are statistically independent, permitting the use of standard statistical formulas (for example, those pertaining to the Student-*t* distribution) for calculation of confidence intervals for the performance metrics of interest. These formulas provide confidence intervals whose width varies inversely as the square root of n (= number of replications), not inversely as n. Therefore, for example, if the width of these intervals needs to be halved to give the client sufficient confidence when making decisions based on the simulation analysis, it is insufficient to double the number of replications. The number of replications must be quadrupled. Furthermore, the analyst must avoid the mistake of making one extremely long run (for example, using the previous numbers, 9600 hours) and mentally dividing it into 20 "replications" of 480 hours each. Such misconstrued replications are not statistically independent – for example, conditions in the system at 955 hours (near the end of one subdivision) and conditions at 965 hours (near the beginning of the next) are very similar, the result of positive correlation. With independence thus foregone, the foundations underpinning the computation of confidence intervals for the performance metrics are therefore severely compromised. Indeed, breaking a "long" run (replication) into pieces can be done, using the technique of batch means and taking care to ensure the batches are as nearly independent as possible (Sanchez 1999).

When only a very few alternatives are to be compared (e.g., a, b, and c), the analyst can reasonably build confidence intervals for all comparisons needed (here, a relative to b, a relative to c, and b relative to c). However, much greater statistical power is available for the typical situation of multiple comparisons on multiple factors. For example, the analyst may need to investigate a situation such as:

A. Conveyor currently in use versus a faster conveyor.
B. Current material-handling equipment versus one more fork truck.
C. Current repair staff level versus one more repair person.

In situations such as this, the analyst can and should use Design of Experiments (DOE). This powerful statistical methodology, using designs such as one- or two-way analysis of variance, a full factorial design, a fractional factorial design, or others, can readily analyze the alternatives collectively. A significant advantage of DOE is its ability to detect interactions. In this example, it may be the case that a faster conveyor, by itself, will yield almost no improvement and an additional fork truck, by itself, will yield almost no improvement – yet making both changes will yield a significant improvement.

ACKNOWLEDGMENTS

The author gratefully acknowledges the help and encouragement of Professors Onur M. Ülgen (PMC and University of Michigan – Dearborn) and Y. Ro (University of Michigan – Dearborn). Furthermore, two anonymous referees have provided valuable and explicit suggestions to improve this paper.

REFERENCES

Brooks, F. P. 1995. *The Mythical Man-Month*, 2nd edition. Boston, Massachusetts: Addison-Wesley.

Cheng, R. C. H. 1993. "Selecting Input Models and Random Variate Generation." In *Proceedings of the 1993 Winter Simulation Conference*, edited by G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, 34-40. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Goldsman, D., and G. Tokol. 2000. "Output Analysis Procedures for Computer Simulations." In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 39-45. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Habenicht, I., and L. Mönch. 2004. "Evaluation of Batching Strategies in a Multi-Product Waferfab by Discrete-Event Simulation." In *Proceedings of the 2004 European Simulation Symposium*, edited by G. Lipovszki and I. Molnár, 23-28.

Jacobs, F. R., W. Berry, D. C. Whybark, and T. Vollmann. 2011. *Manufacturing Planning and Control for Supply Chain Management*. New York, New York: McGraw-Hill.

Kelton, W. D., J. S. Smith, and D. T. Sturrock. 2013. *Simio and Simulation: Modeling, Analysis, Applications*. 3rd edition. Simio LLC.

Khalili, M. H., and F. Zahedi. 2013. "Modeling and Simulation of a Mattress Production Line Using ProModel." In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 2598-2609. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Murty, V., N. A. Kale, R. Trevedi, O. M. Ülgen, and E. J. Williams. 2005. "Simulation Validates Design and Scheduling of a Production Line." In *Proceedings of the 3rd International Industrial Simulation Conference*, edited by J. Krüger, A. Lisounkin, and G. Schreck, 201-205.

Myers, G. J. 1979. *The Art of Software Testing*. New York, New York: John Wiley & Sons.

Rohrer, M. W. 1998. "Simulation of Manufacturing and Material Handling Systems." In *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, edited by J. Banks, 519-545. New York, New York: John Wiley & Sons.

Sadowski, D. A. and M. R. Grabau. 2000. "Tips for Successful Practice of Simulation." In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 26-31. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sanchez, S. M. 1999. "ABC's of Output Analysis." In *Proceedings of the 1999 Winter Simulation Conference*, edited by P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 24-32. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sargent, R. G. 1992. "Validation and Verification of Simulation Models." In *Proceedings of the 1992 Winter Simulation Conference*, edited by J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson, 104-114. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Stansfield, T., R. Massey, and D. Jamison. 2014. "Simulation Can Improve Reality: Get More from the Future." *Industrial Engineer* 46(3): 38-42.

Williams, E. J. 1994. "Downtime Data – Its Collection, Analysis, and Importance." In *Proceedings of the 1994 Winter Simulation Conference*, edited by J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, 1040-1043. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

EDWARD J. WILLIAMS holds bachelor's and master's degrees in mathematics (Michigan State University, 1967; University of Wisconsin, 1968). From 1969 to 1971, he did statistical programming and analysis of biomedical data at Walter Reed Army Hospital, Washington, D.C. He joined Ford Motor Company in 1972, where he worked until retirement in December 2001 as a computer software analyst supporting statistical and simulation software. After retirement from Ford, he joined PMC, Dearborn, Michigan, as a senior simulation analyst. Also, since 1980, he has taught classes at the University of Michigan - Dearborn, including both undergraduate and graduate simulation classes using GPSS/H™, SLAM II™, SIMAN™, ProModel®, SIMUL8®, Arena®, and Simio®. He is a member of the Institute of Industrial Engineers [IIE], the Society for Computer Simulation International [SCS], and the Michigan Simulation Users Group [MSUG]. He has served on the editorial board of the *International Journal of Industrial Engineering – Applications and Practice*. During the last several years, he has given invited plenary addresses on simulation and statistics at conferences in Monterrey, México; İstanbul, Turkey; Genova, Italy; Rīga, Latvia; Jyväskylä, Finland; and Winnipeg, Canada. He served as a co-editor of Proceedings of the International Workshop on Harbour, Maritime and Multimodal Logistics Modelling & Simulation 2003, a conference held in Rīga, Latvia. Likewise, he served the Summer Computer Simulation Conferences of 2004, 2005, and 2006 as Proceedings co-editor. He was the Simulation Applications track co-ordinator for the 2011 Winter Simulation Conference and the 2014 Institute of Industrial Engineers Conference. His email addresses are ewilliams@pmcorp.com and williame@umich.edu.