Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

PERSPECTIVES ON TEACHING SIMULATION IN A COLLEGE OF BUSINESS

Robert M. Saltzman
Theresa M. Roeder

Department of Decision Sciences
San Francisco State University
San Francisco, CA 94132, USA

ABSTRACT

In this paper, we explore the challenges and opportunities we face teaching computer simulation at a business school. While our students tend not to be as technically savvy as most engineering students, at times limiting the technical complexity of what we can cover in our courses, we are able to use simulation as a tool to analyze and discuss business problems in-depth. We explore the differences both between business and non-business simulation courses, as well as those between our graduate and undergraduate courses.

1 INTRODUCTION

In this paper, we explore the challenges and opportunities we face teaching computer simulation in the College of Business at San Francisco State University (SFSU). Our undergraduate course (DS 408) is aimed at business students majoring in Decision Sciences (DS), our graduate course (DS 851) at students pursuing an MBA with an emphasis in Decision Sciences. Both courses run for a full 15-week semester (roughly 45 hours of instruction) and are the only simulation courses available to business students.

While our students tend not to be as technically savvy as most engineering students, at times limiting the technical complexity of what we can cover in our courses, we are able to use simulation as a tool to analyze and discuss business problems in more depth than we are able to with other tools such as Linear Programming.

After briefly reviewing some of the relevant literature in the next section, we describe how we teach our undergraduate and graduate courses in Sections 3 and 4, respectively, indicating how they may differ from those offered by engineering, computer science, or other non-business programs. In Section 5, we compare the content of our two courses before concluding the paper in Section 6.

2 RELEVANT LITERATURE

An early paper by Roberts et al. (1982) describes a panel discussion of how simulation is taught to undergraduates in non-business programs at five different universities. While the panelists have varying approaches, some common themes emerge, *e.g.,* most employ a general purpose simulation language. Courses embedded in engineering departments tend to emphasize engineering applications of simulation, *e.g.,* to issues of industrial design, and often culminate with teams of students solving a realistic problem and writing a report for a client; those embedded within computer science departments may be less application oriented and more theoretical, *e.g.,* they may examine the differences between simulation languages and other programming languages, or the differences between discrete and continuous systems.

978-1-4799-2076-1/13/$31.00 ©2013 IEEE

Another panel discussion reported on by Jacobson et al. (1994) stresses the importance of designing a simulation course so that students who complete it will bring a mix of keys skills to their first post-academic modeling assignment, including not only how to design, build and run models but also how to make formal presentations, write technical reports and communicate clearly with domain experts. Other panel discussions, such as Altiok et al. (2001) and Freimer et al. (2004) address a wide variety of approaches to simulation education, depending on the professor's background and the type of student and program. In this paper, we focus specifically on undergraduate and graduate business students at a business school, similar to Ståhl (2007). Ståhl summarizes his extensive experience teaching simulation to various types of students, from high school to Executive MBA, in a variety of formats, from small course (2-4 hours) to full course modules (30-40 hours). In all settings, Stahl uses a streamlined version of GPSS called WebGPSS, rather than Arena (Kelton, Sadowski, and Swets 2010), because he regards Arena as too complicated.

Standridge (2000) describes his positive experiences introducing undergraduate and graduate engineering students to discrete event simulation via a series of case studies covering issues in manufacturing. By their realistic and relevant nature, cases tend to motivate students to learn the subject matter, while giving them experience with problems that may arise in their jobs. This case-based approach is similar to the one taken in our graduate-level simulation course.

We have found relatively little literature focusing specifically on simulation education in business schools. These types of courses can present unique challenges as business students may not possess the same technological or quantitative skills as their counterparts in engineering programs. Nonetheless, simulation can be a powerful tool for business decision making, and, we feel, can be used to develop students' quantitative and critical thinking skills.

3 UNDERGRADUATE SIMULATION COURSE

3.1 Course Overview and Goals

The undergraduate business program at SFSU does not tend to attract the university's most quantitatively-oriented students. Many of the brightest business students choose the Decision Sciences major, but have not been exposed to programming or more formalized structured thinking. When they enroll in the computer simulation course, they have taken introductory business statistics, though it may have been several years in the past. Approximately half the class are competent users of Excel thanks to other DS courses.

This background has shaped the goals of the course: 1) familiarize students with simulation methodology and analysis, including input and output analysis; 2) introduce students to the fundamentals of structured logical thinking and/or programming; and 3) develop students' oral and written communication skills. We use three software packages over the course of the semester: Microsoft Excel, Sigma (Schruben and Schruben 2003), and Arena.

Because students are familiar with Excel, the course begins with spreadsheet Monte Carlo simulations. In this context, we introduce the ideas of random numbers and distributions, and reestablish the fundamentals of sampling and estimation. During these first two weeks, students see a variety of concrete business and personal applications of simulation, for example, failure time estimation and financial planning.

After students have become familiar with the idea of randomness, they are introduced to the other distinguishing characteristic of discrete-event simulation, time. At this stage of the course, we transition to Sigma as our simulation tool. Here, students start by building simple single-server models. These models are gradually enhanced to include variable-sized arrival and service batches, balking, priority queues, tandem queues, simultaneous resource usage, etc. During this time, the focus is on the numbers of jobs and resources at each stage in the system, not on the specifics of each job or resource. This resource-driven approach (Roeder et al. 2002) has the advantage of abstracting away detail that is not relevant as the fundamentals of the system dynamics and logical thinking are being established. Sigma forces students to develop state change and event conditions "from scratch," furthering the goal of introducing students to programming logic.

It becomes clear that the "pure" resource-driven approach is not appropriate in all instances. As discussed in Roeder and Schruben (2010), both modeling capability and simulation output are limited. We introduce students to the idea of maintaining information on specific jobs and resources. This comes at a large development cost (and, depending on the model size, at a computational cost). Students learn first-hand the tradeoffs between different modeling approaches.

At this point in the semester, we introduce the Arena software suite. There are several reason for this. Students have seen how simulation engines function, but most are highly unlikely to be using software such as Sigma on the job. Rather, they will be using a commercial off-the-shelf package like Arena. In this section of the course, we discuss simulation world views (Kiviat 1969) and how different approaches can have significant implications for modeling and output options. Students also become familiar with the prevalent block-language structure of most DES packages.

The remaining third of the semester is spent on input and output analysis, and special topics such as variance reduction and simulation optimization. In input analysis, we discuss the impact distributional choices can have on simulation outputs, as well as dangers associated with relying too much on historical data. Students collect data for a worksheet assignment and analyze it using Arena's Input Analyzer. In output analysis, we discuss the danger of applying classical statistics on non-iid output data, and common approaches taken to mitigate the problems. We also address initialization and termination bias.

3.2 Pedagogical Details

3.2.1 Simulation Modeling and Analysis

Rather than teach students a single software package, the course tries to expose them to a variety of approaches and tools. Just as there is no single Operations Research tool to solve all problems, there is not a one-size-fits-all simulation tool – even in the more narrow field of discrete event simulation. Since students are unlikely to use the same simulation package they learned in school on the job (if they even remember how to use the package), we feel students are better-served being exposed to a variety of packages. That way, they explore different approaches to modeling, which both deepens their understanding of modeling in general, and of simulation modeling specifically.

The modeling approach taken has a direct impact on the types of things that can be modeled, and on the available output. It can also be a strong determinant of development and execution speed. In many cases, the approach taken will not matter. Nonetheless, awareness of modeling decisions is an important step for the maturation process of a modeler.

A common challenge seen in industry is the desire to include "everything" in a simulation model (Pillai 1990, Roeder 2006). Unfortunately, this may be the exact opposite from what should be done. For example, if a facility that does not yet exist is being modeled, it makes no sense to try to include low-level details about the hypothesized facility in a simulation. Rather, high-level experimentation (and possibly sensitivity analysis) would be more appropriate to gain insights into system dynamics. Students are unlikely to consider this aspect, and succumb to the temptation of high-resolution models.

Because it is not possible to achieve complete fidelity to the real system, it is critical that students explicitly and clearly state any assumptions made during the modeling and analysis of the problem. Especially in simulation, where there *is* the opportunity for much more realism than may be possible in other types of modeling, all parties involved must be clear on what the basis for decision making is.

3.2.2 Programming and Structured Thinking

Because our students are not earning engineering, applied mathematics or statistics degrees, they are unlikely to find themselves in highly technical jobs after graduation. On the other hand, they are receiving degrees in Decision Sciences, and should be expected to be able to conduct reasonable analyses on potentially large amounts of data. Because of the sheer size of data sets, some form of programming or other structured thinking skills are non-negotiable.

This is one of the greatest challenges to our students: As business majors, they are not required to take programming classes. They take two information systems classes, the latter of which includes a project in Microsoft Access. In their DS courses, students become competent users of Excel. While Excel can be a useful tool for analysis, even the ability to create macros or procedures can significantly improve analysis efficiency and reliability. While students implicitly deal with many programming concepts in Excel, they have a surprisingly difficult time translating the ideas from Excel to a more general programming environment.

Basic programming concepts with which the majority of our undergraduate students are unfamiliar at the beginning of their simulation course include:

- Variable types: Because they have worked almost exclusively with spreadsheets, students are not aware of the difference between integers and real numbers. Despite having used arrays and matrices extensively in Excel, students have a difficult time grasping them in the abstract.
- Precedence of statements: The ordering of statements matters. For example, when a new job enters the system, it must be added to the waiting jobs before the condition “minimum batch size~~d~~ available?” can be checked.
- Logical conditions: The combination of true/false conditions using “and” and “or” statements is usually very challenging. Similarly, students often struggle with finding conditions that cover all possible outcomes or states of a system.
- Debugging: Students typically lack the understanding (and perhaps the patience) to single-step through models to find errors. Rather, they prefer to execute the model, then examine the output trace or graphs for “obvious” errors such as negative queues. The latter is a valuable tool, but will only capture certain errors. (It also relies on flexibility in coding to “allow” the model to make mistakes that aid in debugging.)

While it can easily be argued that, for example, it is unnecessary for students to know the finer details of variable typing, we feel that the exercise is perhaps less about defining variables appropriately and more about training students for rigorous thought. Depending on the environment, improperly typed variables can have drastic and, to the uninitiated, sometimes extremely strange consequences. Even if students do not need to program on the job, precise analytical and critical thinking are very important.

3.2.3 Communication

Written and oral communication skills are extremely important in almost any job. In a technical field at a business school, students must learn to present their projects so that non-experts are able to understand the work and the recommendations. Over the course of the semester, students write two homework reports and a final project, and present a worksheet problem to the class. They must also write and present a final project proposal, and submit a progress report. The final project includes a presentation. 65% of the final grade is based on these reports and presentations.

In addition to communicating their work to others, students must communicate with each other. Specifically, each student must provide feedback on the presentations each other student. The feedback is graded on the basis of content and tone; constructive criticism and individualized comments are encouraged. This allows students to practice observation as well as communication skills.

3.3 Student Deliverables

The simulation course meets once a week for 2 hours and 45 minutes, in the computer lab. Roughly half the time is devoted to lecture and sample models or problems; students spend the remaining time on worksheet problems, due the following week. They have the option to work individually or in assigned groups. Two of the worksheets require a 5-10 page report, written in a manner that is understandable to someone unfamiliar with simulation modeling and analysis. Each group must also present one worksheet problem to the class, again understandable to a layperson. The worksheet problems cover modeling problems as outlined in Section 3.1, adding progressively more complexity to basic queueing systems. For example, students begin with a FIFO single-server system which is augmented to include multiple (identical) servers. Jobs then arrive in variable-sized batches, but are served individually. This progresses to fixed batch sizes and finally to variable batch sizes. Simultaneous resource usage is introduced with a worker that loads and unloads a single machine. This model is made significantly more challenging by increasing the number of machines. (This exercise can also lead to an interesting discussion on calculating output statistics and the importance of transparency in communication – should the load/unload time be included in calculating machine utilization?)

Almost 50% of the final grade is based on a group final project. For this, students write and present a project proposal. Each student in the class provides feedback to the other groups: What was the main point of the proposal, should the project be "allowed," and any ideas or comments on the project. Several weeks later, each group submits a progress report. The final project report and presentation are due the day of the scheduled final exam. Each group presents for 10-15 minutes, depending on class size. Both report and presentation are targeted at the layperson.

In addition to giving feedback directly to each group on project proposals, each student fills out a form with feedback for each group presentation/presenter. For worksheet presentations, comments are summarized by the instructor and given to the presenter. For the final project, comments are transcribed (anonymously) and given directly to the presenters. Students are graded on the quality of the feedback they provide each other. Trite or unconstructive feedback is not acceptable; students must be able to evaluate presentations critically and provide personalized feedback. If improvements on the part of the presenter are suggested, the feedback must be given tactfully.

4 MBA SIMULATION COURSE

4.1 Course Overview and Goals

Four major themes are interwoven throughout the course: 1) applications of simulation models; 2) probability and statistics; 3) simulation methodology; and 4) modeling environments.

Like engineering students, MBA students are more interested in applications than in theory. They're anxious to learn both how to use the software and how it is applied in practice. However, unlike a simulation course for engineers (*e.g.,* Standridge 2000), our course focuses almost entirely on applications in the service sector rather than the manufacturing sector, since virtually all of our students already are, or soon will be, working in the services. Among other requirements, each student must write up a summary of a simulation-based case study from the literature (*e.g.,* an Interfaces article) and make a 5-10 minute presentation on it to the class. Combined with the applications presented by the instructor, students are exposed during the semester to 25-30 service sector settings in which simulation has been successfully applied.

Prior to taking this simulation course, students are assumed to have had only one basic course in probability and statistics. Therefore, some lecturing is devoted to reviewing basic concepts and introducing new ones used in simulation modeling, *e.g.,* when selecting input probability distributions and analyzing output. Fortunately, the software that we use (Arena) has extensive capabilities and can perform virtually all of the required statistical work quickly and easily. We tend to emphasize the meaning and correct usage of statistical procedures, downplaying their theoretical underpinnings.

Our course also spends significant time covering simulation methodology. We examine issues in problem formulation, the need to collect and analyze input data, how to verify and validate a model, as well as how to experiment with a model and analyze output from multiple replications. Difficulties and strategies to improve the chances of implementation are tackled toward the end of the course, such as by building front-ends in Excel and using VBA.

The final component of the course, naturally, is the modeling environment itself. Because our MBA students are already familiar with Excel when they reach this course, Excel provides a good environment to get them started learning about many of the fundamental issues of simulation, such as random numbers, sampling, and output variability. The first three weeks of the course are spent in Excel doing Monte Carlo simulations, *e.g.,* of a simple inventory system, of the net present value of five years' of cash flows, etc. While students are engaged by Excel, they also see some of its limitations. The remainder of the course is spent learning how to build discrete event dynamic simulations in Arena. Students use an academic version that they can run on their own laptop computers, which they are encouraged to bring to class. Except for limits on the total number of concurrent entities and number of modules, this version of Arena has the full functionality of the professional grade package. Most students are captivated by Arena's animation capabilities and can envision using it at their jobs.

4.2 Pedagogical Details

The class meets once a week for 2 hours and 45 minutes. Most classes combine model building with coverage of methodological and/or statistical issues, as seen in Table 1. During the Arena portion of the course, students are exposed to almost all of the modules in the Basic Process, Advanced Process, and Advance Transfer panels. By week 11, students submit one-page proposals for their culminating projects, get feedback by week 12, and spend the last four weeks of the course working on them.

4.3 Student Deliverables

All deliverables shown in Table 2 below are done individually, except for the major modeling project, which is done by pairs of students formed by the instructor. Students who succeed on the first seven assignments are likely to create strong final projects, something tangible they can use at work or in job interviews to illustrate their modeling skills. Most students end up working on projects based on either their own or their teammate's current or former job, or another stochastic system with which they've had extensive contact. Saltzman (2006) describes four excellent student projects in detail. Prior to 2009, some students also later extended their class projects into master's theses. (In 2009, our MBA program dropped its thesis requirement.)

Table 1: List of Weekly Modeling Activities in MBA Course

Week	Models built in class	Major Goals / Constructs Demonstrated
1	Birthday Problem; Coin Flipping	Basics of Monte Carlo simulation: random numbers, sampling, output variability, parameterization
2	Bookstore Ordering; Blood Management	Defining performance measures (PMs); Output analysis: CI for the mean. Simple model experiments.
3	Financial Modeling; Customer Loyalty	PM distributions: variance & percentiles. Normal random variables. Sensitivity analysis: 1-way data tables.
4	(none)	System simplification. Process mapping. Input data collection. Arena's input analyzer: goodness-of-fit tests
5	(none)	Limited data options. Random number generation. Simulation languages. Arena/Siman relationship, details.
6	Small Bank	Basic operations: Create, Process, Decide flow modules; running models; examining output. Improving animation.
7	Restaurant	User-defined variables, entity attributes & statistics, *e.g.,* service level. Resource utilization. Animation issues.
8	Emergency Room	Entity types. Sets. Expressions. Separating logic from flow: Station & Route modules. Sequences.
9	Airline Ticketing	Resource sets & selection rules. Scheduled resources. Terminating system output analysis. Comparing systems.
10	Airline Ticketing	Customer abandonment: Search & Remove modules. Model verification.
11	Multi-Queue Systems	Parallel queues & resources: PickStation module. Steady-state system output analysis.
12	Basic Transporter	Transporter & Distance data modules. Leave & Free flow modules. Model validation.
13	Models with Event-based Time Delays	Hold & Signal, and Match flow modules. Automated scenario processing. OptQuest for Arena.
14	Basic Conveyor	Conveyor & Segment data modules. Implementation Issues: Interactivity, Reading/Writing.
15	PERT	Synchronizing entities via Batch flow module. Front-end interfaces with Excel & Visual Basic.

5 COMPARISONS

5.1 Engineering versus Business

The most obvious difference between typical engineering and business students in the United States is the technical skill with which students enter the program. Simulation courses in engineering programs tend to be junior-level courses, at which point students will have had exposure to the general engineering core, including programing. This allows the technical content of the course to progress more quickly, typically leading to more sophisticated models and final projects.

On the other hand, program goals for engineering and business students are different. Most engineering students will try to find jobs in engineering once they have completed their degrees. This may include jobs building simulation models. Business students may be involved in technical analysis, but at a higher level: Rather than needing to be experts at building models, (undergraduate) business students must be educated, critical consumers who are aware of the potentials and drawbacks of simulation as a decision support tool. This also may lead to a focus on different problem types in engineering versus business; for example, engineers may have more problems relating to manufacturing or operations-level simulations. Business students, on the other hand, may focus on higher-level problems or on services.

DS students can serve as an interface between “pure” engineering and “pure” management. In their simulation and broader business courses, significant emphasis is placed on effective group work and communication skills –more so than in typical engineering programs. DS students are trained at explaining technical material in a non-technical manner. They gain a broad, fundamental understanding of quantitative methods, while also spending significant amounts of time learning about all aspects of business (*e.g.*, marketing, finance, and accounting).

Table 2: List of Student Deliverables in MBA Course

No.	Topic	Purpose / Learning goals
1	Birthday Problem	Excel: Basic Monte Carlo simulation. Need to experiment systematically & organize results.
2	Saving for College	Excel: Intermediate Monte Carlo simulation. Multiple input parameters. Sensitivity analysis.
3	Data Collection & Distribution Fitting	Experience the challenges of data collection. Use Arena’s Input Analyzer.
4	Airport Check-In Process	Arena: Basic skills. Build and run a simple model. Observe output variation. Expectations vs. simulation.
5	Department of Motor Vehicles	Arena: Basic modeling with enhanced animation. Impact of cross-training. Statistical significance.
6	Insurance Company Claims Processing	Arena: Intermediate modeling: Sets; Sequences; Animation separated from logical flow modules.
7	Supermarket Shopping and Check-Out	Arena: Intermediate modeling: Balking. Statistical vs. practical differences. Scenario comparison.
8	Article Summary & Class Presentation	Practice summarizing information & writing succinctly. Improve presentation skills.
9	Modeling Project & Class Presentation	Synthesize all aspects of the course in a dynamic Arena model, written report & oral presentation.

5.2 Undergraduate versus MBA

Our undergraduate and MBA simulation courses overlap in many respects. Both spend considerable time covering simulation methodology, including the analysis of input and output. As business courses, both require students to make multiple written reports and oral presentations, and conclude with a major team project incorporating virtually all aspects of simulation methodology. Furthermore, both courses start off by introducing students to simulation in Excel and end in the Arena environment.

However, one major difference is that the undergraduate course makes a substantial foray into Sigma, where students learn about its resource-driven approach to simulation. During this time, they gain insights into the inner workings of the simulation engine and some of the trade-offs that occur among various simulation world-views. The MBA course, by contrast, spends 12 weeks in the entity-driven, animation-focused environment of Arena, trying to develop in students an intermediate level of skill with this package. One reason for this difference in emphasis in that the overriding goal for the MBA class is to equip students with the skills to build realistic and engaging simulation models on the job immediately upon finishing the course. Undergraduates who earn a bachelor’s degree in business, on the other hand, are unlikely to be tasked with building simulation models on the job without significant supervision and/or training; therefore, it less critical that they be deeply familiar with a particular package. In our view, it is more important for our undergraduate DS majors to learn about structured thinking and logical flow, especially since this *may* be the only course where they will really get a feel for any type of programming.

Another difference in the courses is that our undergraduate course, in some ways, is more technical than the MBA course, delving briefly into special topics such as variance reduction and initialization bias. Our MBA course focuses on getting students ready to apply simulation modeling to service sector problems.

6 CONCLUSIONS

We have found little literature focusing specifically on teaching simulation in business schools. We hope that this paper may provide ideas for other instructors facing this challenge. We have, at times, struggled with how to deal with the unusual nature of our students, compared to our own more heavily technical educations. We welcome any thoughts or comments, and hope that this paper may open a dialogue on the topic.

As a result of writing this paper and critically examining our courses, we realize that some changes to each may be warranted. The undergraduate class might be made more useful by including assignments on current applications of simulation modeling in industry. For example, students may read articles in *Interfaces* or *OR/MS Today*, or even in the popular press. Students would also appreciate an introduction to Arena's animation features. Meanwhile, the MBA class could be improved upon by dropping coverage of infrequently used modules, *e.g.,* transporters and conveyors, and devoting more time to subtle but important issues such as problem framing, model boundaries, and choice of performance measures.

ACKNOWLEDGMENTS

We would like to thank our many students over the years.

REFERENCES

Altiok, T., W. D. Kelton, P. L'Ecuyer, B. L. Nelson, B. W. Schmeiser, T. J. Schriber, L. W. Schruben, and J. R. Wilson. 2001. "Various Ways Academics Teach Simulation: Are They All Appropriate?" In *Proceeding of the 2001 Winter Simulation Conference*, Edited by B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 1580-1591. Piscataway, NJ: IEEE.

Freimer, M., L. W. Schruben, T. M. Roeder, C. R. Standridge, C. M. Harmonosky, and I. Ståhl. 2004. "You Are Going To Teach Simulation - Now What? Tips and Strategies." In *Proceedings of the 2004 Winter Simulation Conference*, Edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 2057-2065. Omnipress.

Jacobson, S. H., D. J. Morrice, D. H. Withers, E. Yücesan, and W. D. Kelton. 1994. "Teaching Simulation: A Panel Discussion." In *Proceedings of the 1994 Winter Simulation Conference*, Edited by J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, 1378-1381. Piscataway, NJ: IEEE.

Kelton, W. D., R. P. Sadowski, and N. B. Swets. 2010. *Simulation with Arena*. 5th ed: McGraw-Hill.

Kiviat, P. J. 1969. Digital Computer Simulation: Computer Programming Languages. Santa Monica, CA: The Rand Corporation.

Pillai, D. 1990. "Material Handling Automation for Wafer Fabrication Facilities." In *Proceedings of the Ninth IEEE/CHMT International Electronic Manufacturing Technology Symposium*, Edited by 277-86. IEEE.

Roberts, S. D., J. Banks, J. Kho, U. Pooch, and J. Ramberg. 1982. "Teaching Simulation to Undergraduates." In *Proceedings of the 1982 Winter Simulation Conference*, Edited by Y. W. Chao, O. Madrigal, and H. J. Highland, 706-710. Winter Simulation Conference.

Roeder, T. M. 2006. Simulation: Why Is It So Hard and What Can We Do About It? Intel High Volume Manufacturing Research Committee, July 2006, at Chandler, AZ.

Roeder, T. M., S. A. Fischbein, M. Janakiram, and L. W. Schruben. 2002. "Resource-Driven and Job-Driven Simulations." In *Proceedings of the 2002 International Conference on Modeling and Analysis of Semiconductor Manufacturing*, Edited by 78-83.

Roeder, T. M., and L. W. Schruben. 2010. "Information Models for Queueing System Simulation." *ACM Transactions on Modeling & Computer Simulation* 20 (2).

Saltzman, R. M. 2006. "What Makes a Student Simulation Project Successful?" *California Journal of Operations Management* 4 (1):14-20.

Schruben, D., and L. W. Schruben. 2003. *Graphical Simulation Modeling Using SIGMA*: Custom Simulations.

Ståhl, I. 2007. "Teaching Simulation to Business Students: Summary of 30 Years' Experience." In *Proceedings of the 2007 Winter Simulation Conference*, Edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 2327-2335. Piscataway, NJ: IEEE.

Standridge, C. R. 2000. "Teaching Simulation Using Case Studies." In *Proceedings of the 2000 Winter Simulation Conference*, Edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1630-1634. Piscataway, NJ: IEEE.

AUTHOR BIOGRAPHIES

ROBERT SALTZMAN is Professor of Decision Sciences in the College of Business at San Francisco State University, where he teaches courses in Statistics, Operations Management, Simulation, and Quality Management. He received his Ph.D. in Operations Research from Stanford University, and his B.S. degree in Applied Mathematics from Brown University. Robert's main research interests are in decision making via animated simulation and optimization modeling. His email address is saltzman@sfsu.edu.

THERESA ROEDER is Associate Professor of Decision Sciences at San Francisco State University. She teaches courses in Operations Management, Computer Simulation, Applied Management Science, and Communications for Business Analytics. She earned her Ph.D. in Industrial Engineering and Operations Research from the University of California, Berkeley. Her research interests are in simulation methodology, O.R. education, and health care. Her email address is tmroeder@sfsu.edu.