Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

TEACHING UNDERGRADUATE SIMULATION – 4 QUESTIONS FOR 4 EXPERIENCED INSTRUCTORS

Jeffrey S. Smith

Department of Industrial and Systems Engineering

Auburn University
3301 Shelby Center
Auburn, AL, 37830 USA

Christos Alexopoulos

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
755 Ferst Drive, NW
Atlanta, GA, 30332–0205 USA

Shane G. Henderson

School of ORIE
Cornell University
230 Rhodes Hall
Ithaca, NY, 14853 USA

Lee Schruben

IEOR Department
UC Berkeley
4131 Etcheverry Hall
UC Berkeley, CA, 94720 USA

ABSTRACT

This paper and the corresponding panel session focus on teaching undergraduate industrial engineering/operations research-related simulation courses. The format brings together four experienced instructors to discuss four questions involving the structure and topic outlines of courses, print and software teaching materials used, and general teaching methods/philosophy. The hope is to provide some experience-based teaching information for new and soon-to-be instructors and to generate discussion with the simulation education community.

1 INTRODUCTION

The idea of this paper and the corresponding panel session is to discuss the teaching of undergraduate industrial engineering/operations research-related simulation courses. The mechanism that we've chosen is for four experienced instructors to answer four common questions about their courses, materials, and teaching methods/philosophies. Mixed in with the individual answers, we include brief discussions of our individual experiences (both good and bad) – experience is the best teacher, after all. While not possible in the paper, we also hope that the questions will spur audience participation during the panel session. The paper is organized such that each of the four questions is a major section (Sections 2-5) and each participant's "answer" is a subsection under the question's section. We end with some summary remarks.

2 WHAT IS THE FORMAT OF YOUR COURSE AND WHAT ARE THE MAIN TOPICS THAT YOU COVER?

2.1 Smith

Our course (Simulation) is based on 2 lecture hours + 3 lab hours per week over a 16-week semester. Over the last few years, we have begun to introduce video segments to supplement the lectures and labs. The major topics that we cover include:

1. Basics of Queueing Networks - The students get a good dose of single-server queueing analysis in the prerequisite course in stochastic operations research. We focus primarily on analyzing Jackson Networks and the impact of variability by introducing M/G/1 and comparing various service distributions with the M/M/1 results. We later revisit this concept using simulation models and expand to network analysis involving non-Markovian distributions and relaxed queueing assumptions.
2. Monte Carlo Simulation - We cover the basics of Monte Carlo and work through several example models. We introduce random variate generation from a practical perspective. We use a single-output Monte Carlo model to introduce/re-introduce the concepts of sampling, parameter estimation, and sampling error in the context of simulation modeling. We use Barry Nelson's MORE plot (Nelson, 2008) to discuss the basics of error vs. risk.
3. Discrete-event Simulation – We cover the fundamental concepts of events, system states, event calendars, and sample path generation/analysis. Students have a lab exercise where the manually generate and analyze a sample path for a simple queueing system.
4. Process-oriented simulation and Simio – We discuss the process oriented view point and its implementation in Simio.
5. Input analysis – We have basic coverage of univariate inputs, distribution fitting and parameter estimation, and assessment of fit.
6. Output analysis and experimentation – We focus on assessing and controlling sampling error, basic ranking and selection, and experimentation strategies and practice. Finally, we introduce simulation-based optimization using the OptQuest add-in.

2.2 Alexopoulos

All simulation courses at Georgia Tech (at the undergraduate and Masters levels) are based solely on lectures. During the last three years, the MS course has been offered jointly with an advanced undergraduate course. Until the mid-1990s the undergraduate class consisted of two hourly lectures and a single three-hour lab per week. This approach worked better than the current approach of three instructional hours per week, but the professor was also responsible for teaching the lab without getting any extra credit. The change occurred during the conversion from the quarter system to semesters.

The undergraduate and Masters courses cover all theoretical topics from the classical texts of Banks et al. (2009) and Law (2015), with the exception of design of experiments and variance reduction techniques. The skeleton of both courses consists of the items in Jeff's list above, but the coverage of each topic depends on the class. At this junction, I wish to point out that I spend 60–90 minutes during the instruction of input data analysis with an introduction to kernel density estimation (KDE) for independent data sets. This topic has become more important with the emergence of "big data" with multiple modes. Fitting of KDEs can be handled very easily via a plethora of functions in Matlab, Python, and R, while sampling from the fitted densities can also be done effectively, e.g., via the algorithms in Section 3 of Hörmann and Leydold (2000). Ranking and selection methods are covered using a small set of slides and the "add-ins" in Arena and Simio by Kim and Nelson, while simulation-based optimization is covered "on the fly" using the OptQuest add-in. The undergraduate class covers a large range of modeling topics, including simulation of vehicle systems and conveyors. The emphasis on modeling is very important because of the rigor of our senior design class, where projects often involve a combination of heavy simulation modeling and analysis.

2.3 Henderson

ORIE 4580 Simulation Modeling and Analysis at Cornell University is an introduction to Monte Carlo and Discrete-Event Simulation, with prerequisites sophomore-level probability/statistics and a second course in programming. The class is large, consisting of approximately 200 students, 110 of which are undergraduates and 90 of which are Masters students. This semester-long course entails two 75 minute lectures and a 2- hour recitation (lab) each week. Five of the lectures, that relate to learning how to use a particular discrete-event simulation package, are "flipped." The course can be viewed as one of two halves, with the first half covering the essentials of Monte Carlo simulation and the second half continuing on with discrete-event simulation. The first half begins with a review of probability concepts, proceeds through output analysis for terminating simulations, random-number generation, random-variate generation, stochastic-process generation, input analysis, and concludes with a taste of variance reduction (antithetics). The second half begins with a review of queueing theory, and proceeds through a hand simulation, modeling with a simulation package (where the lectures are flipped), spends some time on comparing alternative systems, and concludes with some discussion of verification, validation, accreditation and project management.

2.4 Schruben

The undergraduate simulation classes I have taught at Cornell and Berkeley have had the same format; 2 to 3 weekly lectures and a lab/recitation with exercises, two exams, and a term product. Both schools have semester-long courses. I only taught a quarter course once (at Florida) and it didn't go well. My overarching goal is to challenge my students to recognize and ask well-posed questions and to identify the assumptions behind the answers. They also learn to assess if each assumption tends to be optimistic or pessimistic, and try to find out, through sensitivity analysis, how much it matters.

Structurally, my courses have been a mix of Art, Science, and Technology:

- **Art:** for about half of my undergraduate course we explore different ways to model dynamics. I believe it is important that my students be able to see how systems change over time from different viewpoints. At a high level: they define a system from the perspectives of the *resident-entities* (the resources or servers, such as an airport boarding gate) and from the perspectives of the *transient entities* (as the airport passengers). These two viewpoints are natural for many industrial engineering systems, such as queueing networks or supply chains. They are mathematically dual perspectives (Little's Law relates time-averages of what servers see (their queues) to count-averages of what matters to each customer (their waiting times)). The students learn how to model very large systems with very small and fast simulations from a resident-entity perspective. Then they see how to augment their models to track the detailed experiences of the transient entities.

  In this modeling "art" part of the course, we use three different types of graphs and a lot of pictures (available on request). For discrete-event simulations we first look at systems as fundamental event relationships using event graphs (Schruben 1983), then as resident-entity activity interactions using stochastic Petri nets (Haas 2002), and finally as transient-entity flow processes with entity flow diagrams using the basic 7 GPSS blocks (Generate, Terminate, Queue, Depart, Advance, Seize Release, and Test); I like GPSS's shapes, they given students visual reinforcement (and makes them appreciate modern software when they see it later). Using these three graphical tools result in different models for the same system, each having different strengths and weaknesses. In teaching modeling Art, pictures are much more important than software for viewing systems from different perspectives and world views. No single paradigm is ideal for all problems they will face in their careers.
- **Science:** this is where we look at modeling randomness (input modeling) and analysis of simulated data (output analysis). The usual material on pseudorandom numbers, random variate generation, and processes are covered with an emphasis on modeling dependency and non-stationarity. We then cover batched means, run initialization and control, variance reduction techniques, and experiential designs with an emphasis on simulation response optimization and sensitivity. One popular excursion is where I show them how to apply cdf inversion to optimize the Newsvendor Problem only knowing the formulas for the areas of a triangle and a rectangle (ask me sometime over a beer.) Other than that, this is pretty dry. To liven things up David Goldsman sent me a VBA program once that was exceptionally well done and entertaining. Sam Savage sent me an excel program with independent

marginal variates whose scatter plot is a happy face. I have tried games and contests to make this less boring; nothing works, except when I toss my 2-headed quarter to show that 3 tosses of a coin is not the same as tossing 3 coins (that is a much deeper lesson than it sounds).

- **Technology**: Just like I teach my students 3 graphical modeling world-views, I introduce them to 3 software packages. For event modeling (I have used Sigma, but am moving to Tao – which I hope to introduce at this conference.) I enable them to build simulations of very large-scale activity models with RTMS from Bio-G.com (with disclaimer, I am a co-Founder). They see how to simulate transient-entity flow models using Simio, Automod, GPSS, Siman, Slam, Flexsim, Arena, and/or Promodel. I have been happy lately with Simio, mainly because it takes no work on my part and little class time. I just point my TA to Jeff Smith's on-line resources (http://jsmith.co/node/26). Once my students learn how to model with event graphs thinking of these graphs in being composed of resident-entity activity *cycles* or process flows *paths* is easy – about ½ hour each with examples.

  The reason I teach my students 3 world views with at least 3 software implementations is that, like everything in life, if one only knows two "things" about something (politics, religion, simulation) they can only see differences; they must know at least three things to see similarities (and more than just one thing to see anything).

For new faculty: the more Art I put into my course, the higher my teaching evaluations. I am not sure whether the students like the Art so much, or they don't like the Science (boring?) or technology (frustrating?). My median course evaluations have varied from a 4-way tie for the third worst in our department to perfect in every category.

3 WHAT BOOKS, PAPERS, AND OTHER LEARNING MATERIALS DO YOU USE AND ARE YOU SATISFIED WITH THE MATERIAL?

3.1 Smith

I use our textbook (Smith et al., 2017) as the primary text for our undergraduate and introductory graduate simulation courses. In addition, I also use a significant amount of supplementary material from Law (2015), Banks et al. (2009) and several WSC papers and related resources that I've collected during my teaching career. Over the last 5 years, I have started using quite a lot of video modules to supplement our lecture and lab sessions. These seem to have made a significant impact – especially as our typical undergraduate class size has grown to 90-110 students. With this many students, I have found that doing any significant modeling in a lecture session is not effective. I've found this same lack of effectiveness regardless of whether we're modeling in Simio, Arena, Excel/@Risk, Python, or Matlab (the programming languages our students learn) – so I'm convinced that this phenomenon is not specific to a language/tool.

In response, my goal has been to move much of the "mechanical" aspects of modeling to the video modules. This lets the students consume (and hopefully learn) the material at their own pace. Ideally, this frees up lecture time to focus on concepts, philosophy, and more general discussion about the topics. My limited experience shows that this strategy works very well *if* the students use the video modules *before* exam/assignment time.

I am generally satisfied with the content of the available material, but it would be nice to have a single book covering this material. As an instructor, I'm looking for that single book that has sufficient coverage of the theoretical aspects *and* practical/tool-based content that I use. As a co-author of a textbook, I thought that this would be easy to create but was totally wrong in this thinking – hence I still use the mix of materials.

3.2 Alexopoulos

We have used the aforementioned text of Banks et al. (2009) for over 30 years, but now the book is aged, with no upcoming revisions in sight. There are texts that combine basic coverage of theoretical concepts with languages/packages, such as Arena and Simio, but in my humble opinion those texts fall a bit short on the coverage of the theoretical topics. Currently, in the undergraduate course I use the text by Smith et al. (2017) and a plethora of slide presentations that are available from a Georgia Tech portal. The Master's course is based on the text of Law (2015) and the Simio text above. In the last few years, I have also used extensively the workbook by Joines and Roberts (2015).

An issue with texts using Simio as the modeling tool was the frequent updates of the software with enhanced object definitions and properties. I should point out that this frequency has decreased during the last year as Simio has matured substantially. Despite these changes, both Simio-based texts are extremely useful and offer a proper coverage of the main modeling concepts and software components. The availability of the instructional videos by Jeff Smith really helps. In the past, my life as an instructor was more stable as I taught GPSS, Arena, and AutoMod, all of which were very mature.

3.3 Henderson

I primarily use a coursepack that I wrote, with some material adapted from notes shared by various luminaries, particularly Peter Glynn and Barry Nelson. I also make Law's "Simulation Modeling and Analysis" recommended reading but not required, and I make available on desk copy at the library many other standard books, which, by the way, are rarely used. I also use a set of labs that have been developed over time. They are a mix of written classroom problems and computing problems.

Students are expected to enter this class having taken sophomore-level probability and statistics, and a second course in programming. For students that are taking the class as undergraduate Operations Research majors at Cornell this background is somewhat recent. For the other students, including Master of Engineering students with undergraduate degrees from other universities, the background is present to a varied extent. Those with weaker probability/statistics backgrounds struggle in the first month of the course. It would be extremely helpful to have a set of, say, 100 multiple-choice questions that students could take to self-assess their preparation for the course, and for those who do not have that background, a tutorial of some form to help them attain the requisite background. This tutorial would be closely related to undergraduate textbooks in probability and statistics, although it would likely be fairly selective. At present I simply point students to a book and suggest a set of chapters to absorb. This solution does not seem ideal.

3.4 Schruben

The textbook by Choi and Kang (2013) has been a godsend for my undergraduate course. I only cover the first Section, since this book is detailed and dense, but it is very clearly written at an easy mathematical level. I can assign reading and exercises from this book without teaching "from it". With this text it is like my students have already taken a course in simulation. Their questions and discussions are at a qualitatively deeper level. I also reference sections in Law's classic text (Law, 2015), which I use in my graduate course. Again, I don't have to lecture from Law's text, and can teach as if my grad students have already had a simulation course as a prerequisite. I have never "followed" a text book, but with a good one I can almost squeeze two courses worth of material into one.

4 WHAT SOFTWARE TOOLS AND PROGRAMMING LANGUAGES DO YOU USE?

4.1 Smith

I use Simio for the discrete-event simulation portion of the course and Excel, Python, and Matlab for the Monte Carlo portion. Our undergraduate students take Matlab- and Python-based programming courses prior to taking simulation. We teach the Python course in the ISE department and actually cover the programmatic aspects of Monte Carlo in that class – this lets us hit the ground running when they get to simulation. This Fall, I'm planning to incorporate @Risk as part of the Monte Carlo and Input Analysis portions of the class.

Several years ago we also did event-oriented simulation using C and/or Java based on the exposition in Law (2015), but I gave that exercise up as the classes grew in size. I continue to think that was a great learning tool and enhanced students' understanding of the discrete-event simulation mechanism, but there was simply too much overhead to do the programming parts with large classes.

4.2 Alexopoulos

I have been using Simio for the last 6+ years. I have also used ExpertFit by Averill Law and Associates for input data analysis during the past 20+ years. We do require knowledge of a scripting language, such as Matlab or Python. Although my students take required courses for these languages, I often discover that their training is inadequate.

During the last two years, I have also adopted the The SIPmath Modeler Tools for Excel from www.probabilitymanagement.org to teach estimation issues related to risk and error in simulation experiments. I use these tools at the onset of all simulation courses, and have found them to be very useful for spreadsheet-based Monte Carlo simulation experiments. Although this fundamental topic is addressed in Simio with the (S)MORE plot proposed by Barry Nelson, the global footprint of Excel and the ability to conduct experiments and observe the impact of randomness on the estimates of risk and error makes this tool invaluable. For systems that can be modeled using spreadsheets, these tools are also useful for teaching estimation of steady-state means and quantiles based on independent replications.

4.3 Henderson

As noted above, students are expected to have had a second course in programming. Such a background is more than is really needed, because any programming is restricted to very small homework problems that can usually be completed in a few lines. One of the reasons we require a second course in programming is simply to be able to rely on an algorithmic way of thinking that is essentially a second pillar to support learning and understanding simulation.

Students learn the basics of Monte Carlo (model logic, multiple replications, confidence intervals) through "raw" spreadsheet modeling, placing a single replication in a single row and "filling down" to obtain multiple replications. Only after those ideas are mastered do we turn to the use of the @Risk spreadsheet plug-in. In the homework I may ask a simple programming task, such as coding the thinning algorithm for generating nonhomogeneous Poisson processes on the line. In the second half of the course on discrete-event simulation we use Simio. The course includes 5 lectures where we learn the basics of Simio modeling. Those lectures are flipped, and held in a computer lab rather than a lecture hall. For each lecture I require students to come to class having already watched a module from Jeff Smith's excellent series of such modules, and I then give them a modeling exercise to work on. Through this period we also continue the recitations (labs) so the students are essentially getting a "double dose" of labs, in place of the usual lecture plus lab format.

4.4 Schruben

As mentioned earlier, I show my students 3 tools from the perspectives of 3 different worldviews so they can see similarities, not just differences. The software I use has changed over the years. I always start with Sigma, which is minimalist, robust, and stable (I started developing it 3 decades ago). It is designed for teaching the fundamentals of all three worldviews, and students can learn everything about event graphs in less than 30 seconds, they like that. I am very proud of RTMS for very large and complex activity-interaction models (I co-Founded Bio-G a decade ago, but did none of the development). I have moved among the many commercial transient-entity flow languages over the years, but lately have been happy using Simio. However, I recommend switching periodically just to keep current - or whenever there is WSC buzz about some radical innovation (which is rarely true). These languages are all pretty similar for teaching purposes, and the advanced details are perishable knowledge – these will change before students will be in a position to use them professionally. I recommend everyone attend the vendor short courses before using any commercial software in their jobs, even if I have just finished teaching it.

5 DISCUSS YOUR BASIC TEACHING PHILOSOPHY, STRATEGY, AND METHODS

5.1 Smith

Simulation requires skills in the general areas of (1) probability and statistics; (2) programming; and (3) modeling. Early in my teaching career, I assumed that students gained the requisite knowledge in each of these areas through the prerequisite courses (two semesters of probability and statistics, two semesters of programming, two semesters of OR, for our students) and that I would just tie these things together under the umbrella of simulation. What became apparent to me over the first few years (most unfortunate that it took me this long!) was that students required significant reinforcement/review of "prerequisite material" before they could productively apply these general topics/tools to simulation. As a result, my classes now cover approximately 1/3 less "new" simulation-related material than they did early in my career. From a teaching philosophy perspective, this simply means that I spend much time and effort on the *integration* of the three basic skills and this often requires going back to basic material that students have already seen in prerequisite courses so that we can view these as an integrated "whole" in the form of simulation.

As mentioned previously, I have not found it especially effective to lead students through modeling/programming exercises in a "follow-me" fashion in a computer lab. Students' skills and comfort levels when it comes to computer applications/programming vary drastically (and often not relative to their GPA). As such, regardless of the pace that I go, 1/3 to 1/2 of the class will be either bored or lost in this environment. Instead, I have found it more effective to combine focused lectures with video modules on the same topics. As an example, I use this method for the following three fairly unrelated topics:

1. Using a Python module (that I provide) for "solving" Jackson Networks;
2. Developing and applying a standard Confidence Interval; and
3. Developing a Simio model of a tandem queueing system.

For each of these, I present the material in a lecture and tell the students: "put your pencils down, shut your notebook computers, and just watch – there is a video module of me doing the *exact same development* that you can view and re-view at your own pace later." The idea is that they can focus on the concepts and big picture first, knowing that they won't lose access to the details that they will need for exams and/or to assignments. I currently have 8-10 similar lecture+video combinations that I use through the semester (in addition to several video-based lab assignments). While I have not done a controlled experiment, anecdotal evidence suggests that my students perform significantly better when I use this method than they do when they try to take detailed notes and/or follow me on the computer. I continue to develop these modules and all are freely available at http://jsmith.co/node/26 (note that I was surprised and continue to be little embarrassed that some of these modules are prominently mentioned by my fellow panelists in this paper – that was definitely not my indent when I initiated the panel!).

Finally, I have found that semester projects are very important and can definitely crystalize all of the simulation-related topics that the students learn over the semester. I have tried several approaches from using the annual Arena and Simio student competitions, to using case-studies developed from consulting projects, to letting students choose their own project topics and all have proven valuable (although students often find choosing their own projects more difficult than they anticipate). As our classes sizes have grown, I've switched from live "final presentations" to video presentations uploaded to YouTube. While this is a different experience than students get presenting in front of a live audience including their peers, I have not found a practical way to have live, in-class presentations once there are more than 10-15 project groups in a given class.

5.2 Alexopoulos

To explain my philosophy, I have to discuss my background and training in simulation. I took a graduate course from George Fishman in 1984. The class required building discrete-event models using Simscript II.5, and the *only* model we built was the famous African Port model that still appears in the text of Law (2015). We built two variants, the first was based the event-scheduling mechanism and the second used the process interaction paradigm. To understand the differences, we had to report compilation and linkage times on the mainframe computer. The fact that we used the same language made us appreciate the fundamental differences between the two modeling paradigms. We analyzed the simulation output using subroutines from Fishman's (1978) text: the first used the regenerative approach, while the second used a batch-means algorithm. George graded the handwritten reports very carefully and took points off for failure to prioritize the list of findings and recommendations (of course, he could do this because the class consisted of only 12 students). George emphasized his preference of a long run over multiple short replications for estimating steady-state means and quantiles. Overall, his teaching and academic advice shaped my philosophy.

When I arrived at Georgia Tech, I found that Jerry Banks, Dave Goldsman, and Jim Swain were using GPSS/PC for undergraduate instruction and the Simlib package from the text of Law and Kelton (1982) for graduate instruction. I recall studying Thom Schriber's (1974) "red book" and that GPSS/PC allowed "animation" by highlighting active GPSS blocks—this could be done using portable PCs and three-beam projectors, but I can ascertain that such cursory animation allowed the students to appreciate entity movement. On the other hand, the Simlib package was based on the event-scheduling mechanism and was written in Fortran (actually one of my students rewrote the package in C before the C code appeared in the third edition of the text). We switched to GPSS/H in the early 1990s (without using Proof Animation) until the late 1990s when we adopted Arena. I used Arena in both undergraduate and MS courses until about 2008, when I switched to AutoMod for about two years. I found AutoMod to be very appealing because of the embedded simulation language, its excellent graphical interface, its ability to model warehousing systems with appropriate amount of detail, and the existence of an introductory text by Jerry Banks, which is still included in the installation. In January 2011, I made the last switch to Simio, and I have remained "loyal" to it until now.

My instructional mix between modeling and theory has remained quite steady throughout my career, with two changes in the last three years: (a) Early use of spreadsheets to illustrate the concepts of error and risk. I have uploaded the paper of Barry Nelson from the 2008 Proceedings of WSC on my portal site, and I repeatedly test students on their ability to interpret histograms, empirical distribution functions, and confidence intervals for means and quantiles. (b) Increased emphasis on input data analysis and distribution fitting, and the effects of "misfitting" on the quality of the simulation output. I find that the perfect balance between simulation modeling and analysis is a real challenge! The ability to conduct a solid statistical analysis of simulation data is a powerful tool for operations research analysts and industrial engineers; in my humble opinion, this is what separates them from computer scientists who can deliver excellent codes/models, but have rather limited statistical capabilities.

I find teaching software without a lab session a great challenge. To overcome the time constraints in a single class session, I usually upload an intermediate model to the class portal, ask the students to study the relevant portion of text (and probably ahead of it), and then proceed with building the remainder of the full model in class. The aforementioned challenge is partially due to the extensive ownership of Apple laptops by students; this typically necessitates logging to a virtual desktop and using an Apple keyboard. I understand that this is not equivalent to lab instruction, but this is the best I can do with our large class sizes ranging roughly between 40 to 80 per section.

I would like to finish with the recent habit of the students to study only electronic material. I urge them to purchase hard copies of the texts—most not only ignore my advice, but they don't even save the electronic files locally. I guess things are not going to get better in the foreseeable future, and we all have to adopt.

5.3 Henderson

These thoughts are perhaps best presented as bullets:

- Students need a strong understanding of probability and statistics to be effective simulation users. Ideally (from my perspective) they would learn both simulation and probability/statistics hand in hand, but with the curriculum design at Cornell this is not possible. My course heavily emphasizes these aspects.
- Don't build models in lectures. It is very difficult for students to see what is going on in the modeling, and many of the stronger students quickly lose interest. This is what labs are for. I used to build models in lectures, learned quickly that it wasn't an ideal way to proceed, but a range of factors including inertia meant that I didn't redesign that part of the course for some time.
- For each new concept, intuition comes first, and mathematics (or whatever background is needed) comes second. For example, generating a random variable with distribution function F by inversion is a very intuitive notion involving stretching or compacting a uniform random variable.
- Try to reduce the workload towards the end of the course – think of this as "antithetic" to the approach in other classes! To that end, we discuss VV&A, and project management, in the last lecture or two, and there is no homework on that material. I obtained Deb Sadowski's slides from a Winter Simulation Conference tutorial on project management, and use those verbatim, with my own commentary. Also, covering this material after students have participated on a non-trivial team project converts their reaction from "this is obvious" to "I wish we had taken minutes at meetings!"
- The course includes a team-based project that requires model building, data analysis and report writing. Grading reports is a nightmare, but the effort is, I think, worthwhile. Alumni often remind me of projects we have done in the past that resonated with them (or horrified them). I've never figured out how to avoid a project hand-in date that is essentially the end of the semester (see the previous point on workload). Any ideas?

5.4 Schruben

I used to teach that a "model" was a noun, a tool you build and use for analysis. Mainly learning-by-doing in my consulting work, I now teach "to model" as a verb, an active verb with a direct object. We model every aspect of our lives to structure our thinking and communicate ideas. Therefore my course outline is no longer linear, following the textbook "steps in a simulation study". Now my course outline, like my professional work, is composed of concurrently interacting cycles illustrated in Fig. 1. (Note there is a "Start" but no "Finish".) I find this works well in consulting. It is a joy to watch my students mature from simple linear thinkers to creative artists.

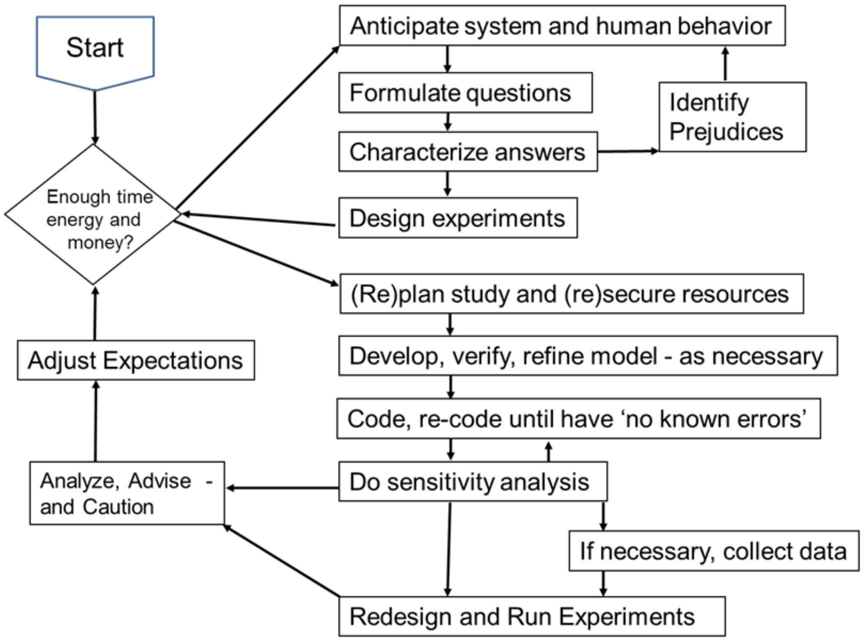
Start
Enough time energy and money?
Anticipate system and human behavior
Formulate questions
Characterize answers
Identify Prejudices
Design experiments
(Re)plan study and (re)secure resources
Develop, verify, refine model - as necessary
Code, re-code until have 'no known errors'
Do sensitivity analysis
If necessary, collect data
Redesign and Run Experiments
Analyze, Advise - and Caution
Adjust Expectations

Figure 1: Modeling (and course outline).

5.4.1 Good Ideas

- **Term Products:** For the past decade I have required that teams of students build a simulation product. Changing the middle two letters from term pro**je**cts to term pro**du**cts has made a huge difference. I strongly recommend it. Their products are of their own creation, and are designed to solve a generic class of problems with an identifiable market. Their final product presentations are YouTube videos that include a product rationale, a demonstration, a market analysis, and a sales pitch. Search (YouTube: Berkeley Simulation IEOR131) for some examples. Students have used these videos on their CVs and credited them with landing jobs; a few have even earned students some money. The first lesson I learned from replacing product reports with videos is to limit the length to 5 minutes! They also turn a user's manual with their product. A mistake I made four years ago was to have teams compete in designing the same simulation product. The competition was fierce, and some of them still won't talk to each other – or to me since I picked the winners.
- **Advanced Placement:** Lecture attendance is not required at Berkeley, I developed a trick that assures high attendance. I give short, in-class "Advanced Placement" (AP) questions without notice. Student's AP scores are added to their next exam score with a separate curve, and they get their best exam grade. Since AP cannot lower their grade, attendance is still optional. My lecture attendance is almost perfect (students even email me excuses when they miss a lecture!). I even have students ask for more AP if I don't offer these for several lectures. Hint: if you ever try this, do NOT call them pop quizzes. Students resent pop quizzes, but they love AP (after I rebranded, my teaching evaluations went up considerably).
- **Creating your own simulation software:** This is a good idea that I don't necessarily recommend. It creates a deep appreciation of commercial software, and you can make money if you can do something of value nobody else can (like build simulation engines in DOS, or 16-bit Windows, or in an internet browser!). However, as Alan Pritsker and Dennis Pegden both told me, you can't both run a successful software company and be a successful college professor, I chose the latter, but still like low-level software development.

5.4.2 Bad ideas (they seemed good at the time):

- **The Million Point Exam**: I got tired of students begging for partial credit. They felt great getting 1000 extra points until they got half-way down the hallway. The resentment never goes away.
- **Peer-grading:** when TA budgets were cut at Cornell, I had randomly paired students grade each other's homework blind, and then grade the graders blind (using course ID's not names – a good idea). Nobody graded anyone who graded them to avoid collusion. In the course evaluations I asked students if they think they graded fairly (all did) and were graded fairly (none were). That year was the low-water mark in my teaching evaluations.
- **Exam questions with negative points**: I once gave a True/False final with equal negative value for wrong choices. A student who missed the exam beat the average.
- **Writing software emulators:** The Sigma download (www.sigmawiki.com) includes a GPSS toolkit, an Arena toolkit, a SIMAN toolkit, Simio and Petri Nets emulators as well as PERT Chart and continuous time simulation examples. The tech support is horrid. If you ever try these, do NOT e-mail me asking for new features.

6 CONCLUSIONS

Most Industrial Engineering and many quantitative business programs offer courses in simulation and teaching methods for these courses are varied. This paper (and the conference panel) focused on the methods, experiences, and methods/philosophies of four experienced instructors. While we don't claim to have perfect answers to all questions or optimal methodologies, we hope that other instructors and future instructors will find the some of the information useful.

REFERENCES

Banks, J. J. S. Carson II, B. L. Nelson, and D. M. Nicol. 2009. *Discrete-Event System Simulation*, 5th edition. New York: Pearson.

Choi, B. K., and D. Kang. 2013. *Modeling and Simulation of Discrete Event Systems*. New York: Wiley.

Fishman, G. S. 1978. *Principles of Discrete-Event Simulation*. New York: John Wiley and Sons.

Haas, P. J., (2002) *Stochastic Petri Nets Modelling, Stability, Simulation*. London: Springer.

Hörmann, W. and J. Leydold. 2000. "Automatic Random Variate Generation for Simulation Input," in *Proceedings of the 2000 Winter Simulation Conference*, edited by P.A. Fishwick, K. Kang, J.A. Joines, and R.R. Barton, 675–682, Piscataway, New Jersey, Institute of Electrical and Electronics Engineers, Inc.

Joines, J. A. and S. D. Roberts. 2015. *Simulation Modeling with SIMIO: A Workbook*, 4th edition. Sewickley, PA: Simo LLC.

Law, A. M. 2015. *Simulation Modeling and Analysis*, 5th edition. New York: McGraw-Hill.

Law, A. M., and W. D. Kelton. 1982. *Simulation Modeling and Analysis*. New York: McGraw-Hill.

Nelson, B. L. 2008. "The MORE Plot: Displaying Measures of Risk & Error From Simulation Output," *Proceedings of the 2008 Winter Simulation Conference*, edited by T. Jefferson, J. Fowler, S. Mason, R. Hill, L. Moench, and O. Rose, 413–416, Piscataway, New Jersey, Institute of Electrical and Electronics Engineers, Inc.

Schriber, T. J. 1974. *Simulation Using GPSS*. New York: John Wiley and Sons.

Schruben, L. 1983. Simulation Modeling With Event Graphs. *Commun. ACM* 26:11 pp. 957–963.

Smith, J. S., D. T. Sturrock, and W. D. Kelton. 2017. *Simio and Simulation: Modeling, Analysis, Applications,* 4th edition. Sewickley, PA: Simio LLC.

AUTHOR BIOGRAPHIES

JEFFREY S. SMITH is the Joe W. Forehand Professor of Industrial and Systems Engineering at Auburn University. He has served as the WSC Business Chair (2010) and General Chair (2004) and is currently on the WSC Board of Directors. He has a BIE from Auburn University and a MS and PhD (both in Industrial Engineering) from Penn State University. His email and web addresses are: jsmith@auburn.edu and http://jsmith.co.

CHRISTOS ALEXOPOULOS is a professor in the H. Milton Stewart School of Industrial and Systems Engineering at the Georgia Institute of Technology. His research interests are in the areas of simulation, statistics, and optimization of stochastic systems. He is a member of INFORMS and an active participant in the Winter Simulation Conference, having been Proceedings Co-Editor in 1995, Associate Program Chair in 2006, and a member of the Board of Directors between 2008 and 2016. He is also an Associate Editor of the ACM Transactions on Modeling and Computer Simulation. His e-mail and web addresses are christos@gatech.edu and http://www.isye.gatech.edu/~christos.

SHANE HENDERSON is a professor in the School of Operations Research and Information Engineering at Cornell University. His research interests include discrete-event simulation and simulation optimization, and he has worked for some time with emergency services and bike sharing applications. He co-edited the Proceedings of the 2007 Winter Simulation Conference. His email and web addresses are: sgh9@cornell.edu and http://people.orie.cornell.edu/~shane.

LEE SCHRUBEN is currently a professor at Berkeley, after spending 23 years at Cornell, and a year each at Purdue and the University of Florida. He has taught simulation for over 40 years and is a co-Founder of the Bioproduction Group (Bio-G.com) and helped design the Sigma and Tao teaching software. He is particularly pleased to be in this WSC Session Chaired by Jeff Smith. Jeff's on-line Simio resources (http://jsmith.co/node/26) have allowed him to retire after this term with a clear conscience. His e-mail address is still LeeS@Berkeley.edu.