Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

SIMULATION EDUCATION IN NON-SIMULATION COURSES

Roger McHaney

Management Information Systems
Kansas State University
3091 Business Building
Manhattan, KS 66506, USA

ABSTRACT

In many curricula and degree programs, simulation courses are not required, but these tools and techniques could be beneficial to students preparing for a variety of careers. The current paper describes two examples of simulation education embedded into broader course topics from a development perspective. The examples, from courses generally described in this paper as Cloud Computing and Big Data, offer a recommended approach for exposing students to practical uses of simulation. In the first example, we use simulation techniques to develop a web service emulation response database within a cloud computing environment for software testing. In the second example, simulation techniques provide an approach to generate data sets for learning data analytics techniques.

1 INTRODUCTION

Simulation is a widely used technique with application in an amazing array of situations (Robinson 2014). Specific tools which facilitate the simulation process are used in engineering, business, healthcare and other fields to solve problems, illuminate nebulous situations, and help decision-makers envision consequences of various actions (McHaney 2009). Many excellent university-level simulation courses exist and are required in various degree programs in areas such as industrial engineering, computer science, operations research, business, and operations management (Chwif et al. 2001; Jacobson et al. 1994). However, not all students that can benefit from simulation are exposed to it in classroom settings. To this point, Stahl (2000, p. 1602) states, "…it is surprising that most business schools and quite a few engineering schools do not give their students any substantial amount of teaching in simulation. In the opinion of many experts, simulation is far away from being as broadly used and taught, as it should rightfully be." In 2017, Greenwood emphasized the need for broadening simulation education in disciplines where its use should be a de facto standard, but too often is not (Greenwood 2017).

This lack of coverage in various disciplines appears to result from assorted pressures, not the least of which is the escalating cost of higher education (Ginder et al. 2017). Other, more direct reasons could include current trends such as one in business schools where credit hour requirements have been reduced. Other reasons are articulated by Stahl (2000) and include: (1) widespread simulation usage requires that teachers have simulation technology knowledge; (2) the cost of learning simulation may be prohibitive or at least substantial; and, (3) student time is a scarce resource in very crowded curricula. In addition, having access to simulation software can be costly and require university computing resource support.

In spite of the constraints, Chwif et al. (2001) report that "[o]ver the past years, there has been a growth in simulation courses both at undergraduate and postgraduate levels." And, in many instances, simulation techniques become part of a portfolio of tools taught within the constructs of a broader class. This approach is not new. As early as 1958, researchers described the use of simulation techniques in operations research (Harling 1958) and simulation usage became standard in the context of many offerings in engineering, business, and sciences.

In recent years, the use of simulations in the classrooms, as opposed to the teaching of simulation techniques has grown. While this idea has been in practice for some time (McGuire 1976), recent technology improvements and new pedagogical approaches have encouraged educators to utilize simulation in the classroom as a mainstay of learning. In fact, classroom simulation use can be found in many disciplines including healthcare and nursing (Shin et al. 2015), emergency response (Lateef 2010), aviation (Jentsch and Curtis 2017), business (McHaney et al. 2018), and many more (Katsaliaki and Mustafee 2015). This use has become more popular in light of constructivist learning pedagogies which encourage teachers to use tools such as simulation to provide richer and more meaningful learning experiences (McHaney 2011).

While many students are exposed to simulations in the classroom, we believe it remains important that simulation development techniques be understood and utilized. Using a simulation or a pre-developed simulator is far different than creating one. As suggested by Stahl (2000), far more students can benefit from simulation use than currently take classes in that area. We agree and covertly have worked to encourage students to use simulation as a quantitative tool to facilitate decision-making or problem solving within the context of their discipline. We provide 2 examples in this paper.

The first example, from a course generally described in this paper as *Cloud Computing*, uses simulation techniques to develop a web service emulation response database within a cloud computing environment for software testing. A simulation creates a script for testing that can be reused or easily modified by students to improve software testing. Simulation becomes a tool used in a practical way to enable their larger efforts in software development, and it leverages tools already familiar in their educational experiences (e.g. Use Case Diagrams). The second example comes from a course generally described as *Big Data*. In this example, simulation techniques provide an approach to generate data sets for learning data analytics techniques and understanding the stochastic nature of model outputs. The idea is to build relationships into the data and then hone skills that ensure the expected outcomes can be visualized within a controlled environment.

2 EXAMPLE 1: CLOUD COMPUTING

2.1 Scenario

Our first example is based on a cloud computing system that relies on a web service to return responses to a request. Cloud computing and web services are not the same thing, however, they can depend on each other in particular circumstances. In this classroom scenario, students are required to have a cloud-based database system constructed and running on Microsoft Azure Cloud. The database needs to provide responses based on a sequence of input parameters passed from student-developed, web-based, mobile, or console software systems.

A system such as this can be tested in several different ways. One method is to request a set of information from the database and compare results to expectations to determine if the system works appropriately. This approach is fine for small systems but may not be comprehensive enough to catch all potential problems. Another testing method involves creating a script meant to test typical requests. This script can be expanded to include requests that might be unusual. This can be done through use of logic statements to ensure a variety of cases exist. Developing a script can be time consuming and may not include all anticipated extremes or expected requests. Likewise, it may fail to take the sequence of requests into account. In other words, in some instances, the order in which requests occur could impact the system. Simulation can add stochastic and comprehensive elements that might otherwise be missing.

2.2 Classroom Use of Simulation for Cloud Systems

A number of methods for generating a script can be used. Creating a simulation with input distributions related to request parameters falls into this category. Using this approach offers a natural classroom moment to introduce simulation techniques to a group of tech-savvy students that can integrate this tool into their portfolio of decision-support techniques. For example, students are provided with input data that suggests a certain percent of requests to a Web Service need information related to particular country. A discrete input distribution can be created to mirror the expectations. Table 1 provides a sample composite of a portion of the dataset used by students.

Table 1: Composite data for input distribution.

ID #	Country	# Car Sales
5	United Kingdom	215
6	USA	166
1	France	40
4	Switzerland	26
3	Spain	6
2	Germany	4

Students are instructed to utilize the raw version of this data to create an input distribution that will incorporated into their simulation. One tool at their disposal is EasyFit Professional from MathWave Technologies. They input the data to discover an appropriate distribution. In the case of this dataset, a Binomial distribution with parameters as shown in Figure 1 was recommended. The result is coded into an AnyLogic statement (e.g. Binomial(0.60385,8)) to be used in the model. Logic statements route entities to further assignments based on values from example data provided by the instructor. In this example, the ID # becomes a value included in the generated data set.

#	Distribution	Parameters
1	Binomial	n=8 p=0.60385
2	D. Uniform	a=3 b=7
3	Geometric	p=0.16938
4	Logarithmic	θ=0.92811
5	Poisson	λ=4.9037
6	Bernoulli	No fit (data max > 1)
7	Hypergeometric	No fit
8	Neg. Binomial	No fit

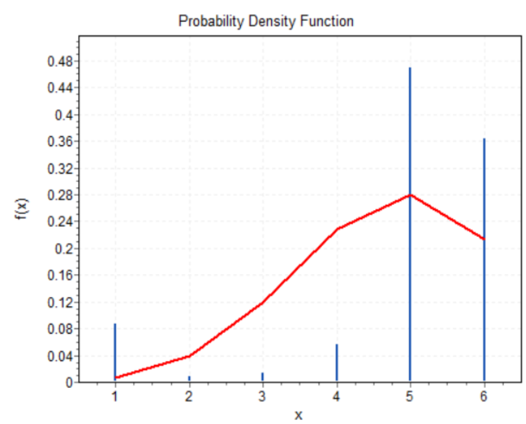

Figure 1: Binomial distribution selected by students using EasyFit.

2.3 Example Simulation for Request Generation

Figure 2 provides a conceptual overview of simulation's integration into the classroom project that requires students to develop a system that sends a structured query language (SQL) query command to a Microsoft Azure Cloud database system. For this assignment, students design a database, build it in a cloud-based environment, and populate it with data. To ensure the system operates as expected, test scripts are created using a programming language in a software development environment. The test scripts pull data sequentially from a list of values generated by the simulation. We used the Personal Learning Edition of AnyLogic, which is free to students, to generate the list.

In the current example, Microsoft Visual Studio was used with the students' choice of either Visual Basic, ASP.net, or C# as a programming language to build the test script program. Other languages such as Java or Python also could be used. Student-generated scripts connect to the Azure cloud system and request data based on a variety of input queries. The students use simulation to generate a sequence of request parameters written to a spreadsheet file. The parameter combinations are generated using a relatively simplistic model depicting a variety of users accessing the database system. A Use Case Diagram describes the types of users for the system and provides a general framework for the model. An example is shown in Figure 3.

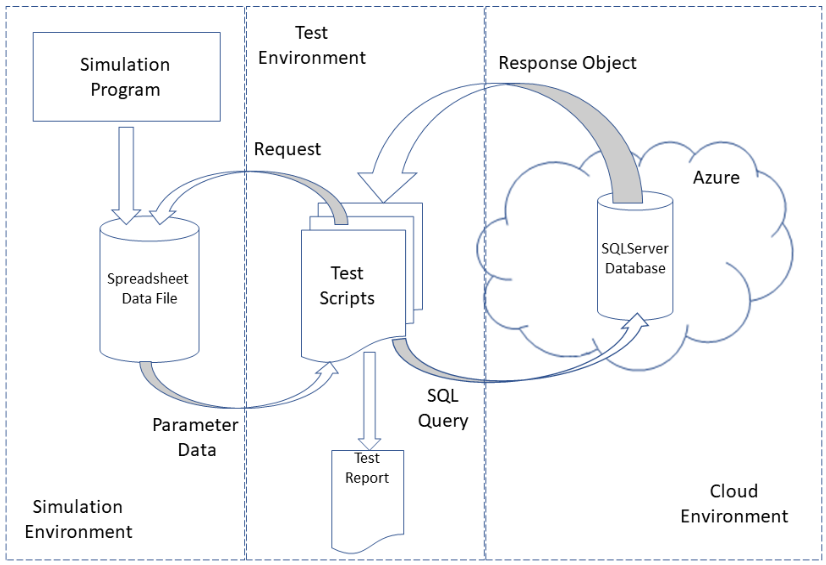

Figure 2: Conceptual approach for simulation use in teaching cloud concepts.

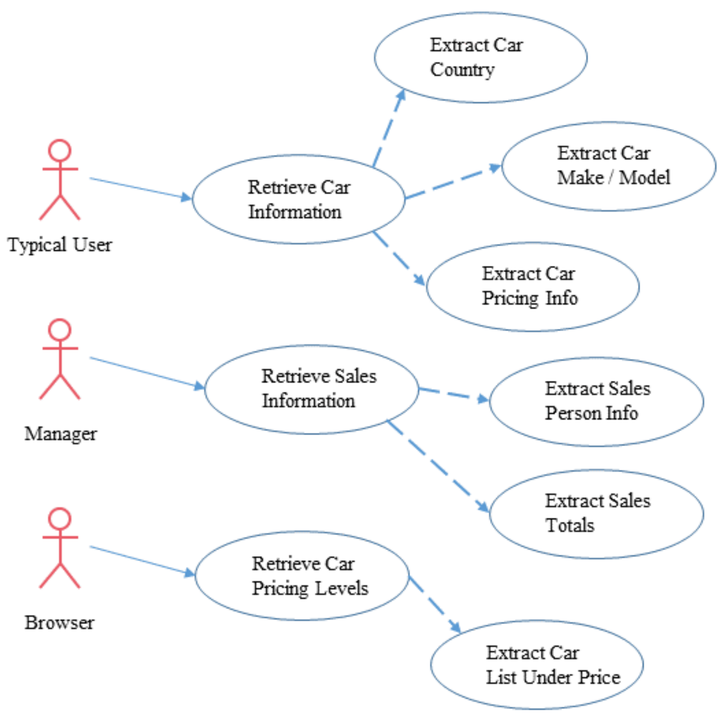

Figure 3: Example use case diagram used to develop conceptual model.

In the model, database users reach a block which represents their request to retrieve data. The model derives parameters based on a series of input distributions that represent characteristics of what the users wish to obtain. In our current example, this could mean the simulation returns the ID # for the *Country of Car Manufacture* based on an input distribution. Likewise, a requested parameter in subsequent blocks could rely on the value held in the first parameter and be populated with appropriately, related information (e.g. Make, Model, Car Sales Price, et cetera). Logic statements would help ensure the correct combination of values is generated. Logic in the simulation constructs a realistic parameter response to be included in the generated test script. In AnyLogic, this means writing values to an output spreadsheet file using code attached to a block such as:

```
ModelOutput.SetCellValue(Agent.CarCountry, 1, SpreadsheetCounter, 2);
```

In this example, students learned to conceptualize a model, derive input data distributions, construct logic sequences for the model, and create stochastic output files. Although the use of simulation is very basic, students come to understand the process from a beginner's perspective. This provides a point from which more complex simulation work becomes possible.

3 EXAMPLE 2: BIG DATA CREATION

3.1 Scenario

Our second example relates to Big Data courses or classes using data visualization. This example uses simulation techniques to generate a data set that can be used in storytelling exercises by students. Again, students experience simulation and can incorporate it into their toolkit of techniques to be used as future decision-makers. This is important because many students taking coursework in business analytics, data science or big data do not specifically take a simulation course but its benefit cannot be overstated.

Of course, other methods exist for generating sample data sets but the simulation approach accomplishes three objectives. One, is to expose students to simulation techniques as described earlier. The second objective permits generation of large data sets with built-in randomness having somewhat unpredictable characteristics. The third advantage relates to storytelling with the data. The students create the simulation to generate data and in doing so, construct a scenario that correlates with the storytelling aspect of big data. Unlike the real world, students know the big picture and then learn how their efforts aid in the 'rediscovery' what they have created through model output analysis. The analysis efforts ensure students recognize the stochastic nature of simulation.

3.2 Classroom Use of Simulation for Big Data

The scenario for this use of simulation involves creating entities that represent a real world element to be analyzed. These entities may be people, processes, a manufacturing assembly or any other object of interest. This 'data' entity moves through a sequence of blocks that instantiate its parameters based on input distributions meant to describe the ranges of data. Like our first example, students come to understand the importance of input data and describe it via well-fit distributions. Ranges of data are fit to distributions, combined with business (or assignment) rules with a series of logic statements, and then used to create output sets housed in data files.

Data ranges can be specified and used in ways that match expected real world values. Data analysis tools can be taught using the resulting data items which have been written into an output file. Figure 4 provides a conceptual framework.

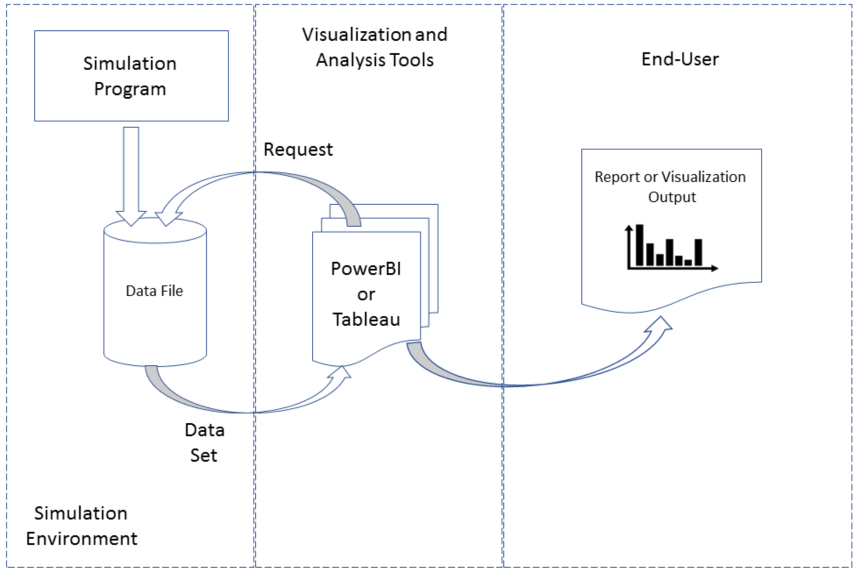

Figure 4: Conceptual framework for big data simulation use.

3.3 Example Simulation for Big Data

For this assignment, students design a model, build it in a simulation language such as AnyLogic, and use it to generate data. A series of input distributions are assigned in a series of block statements. Logic is used to determine an entity's path through the model and this impacts the contents of assigned parameters. For instance, patients entering an emergency room are assigned an illness condition, gender, age, and other characteristics based on the professor's guidelines.

In the current example, Microsoft PowerBI was used to visualize the model's output. Data generated by the simulation provides material for student experimentation. Value combinations are generated using a relatively simplistic model depicting various possible combinations of data based on input distributions that are either assigned by the professor or derived from smaller datasets. Figure 5 shows a typical visualization.

4 SIMULATION CONCEPTS TAUGHT WITH THIS APPROACH

The examples described in this paper are relatively simple simulation projects. The goal was not to create simulation experts but instead create awareness of simulation techniques and demonstrate how these can add value to activities considered central to particular fields of study. In addition to limited simulation development experience, students receive resources enabling them to go further with modeling. In particular, it is important to reinforce concepts related to the stochastic nature of outputs and how results need to be considered through careful analysis.

Challenges exist with rapid introduction of simulation concepts. In particular, students (particularly those with programming backgrounds) tend to jump into model construction without considering the nature of their development. Reflection, regarding potential ranges of model outputs over simulated time, should be encouraged to ensure students recognize they have generated a subset of potential results.

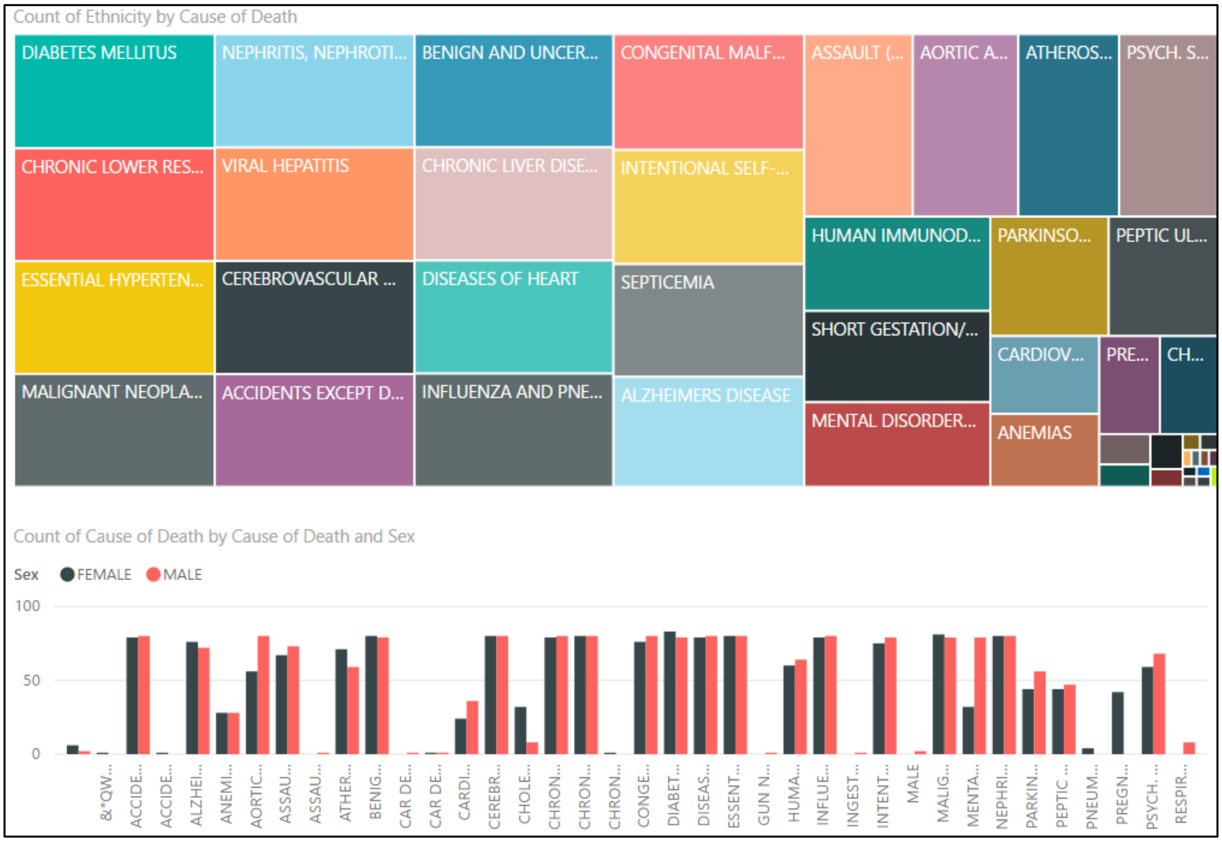

Figure 5: Typical visualization.

The notion of simulations being developed by non-experts is consistent with current trends in end-user computing (Mørch et al. 2017). The idea that authoring tools and agile methods are pushed down into the ranks of users works well using the approach described in this paper. Likewise, conceptual modeling techniques can help inform the novice simulation user of ways to better understand a system (Robinson 2008, van der Zee et al. 2010). Lieberman et al. (2006, p. 1) argue that "the goal of interactive systems and services will evolve from just making systems easy to use (even though that goal has not yet been completely achieved) to making systems that are easy to develop by end users." They justify this statement by suggesting that people are familiar with computer interfaces and will be able to use tools that abstract the process from the programming code level. This low-code, no-code approach is already available in many simulation languages. The data artifacts created using AnyLogic described in this paper are examples. Students do not need to be programmers nor simulation experts. Instead, they work to build a model using available drag and drop features and configuration. This is the world these students will inhabit in their careers.

In short, covert teaching of simulation offers a method to enrich courses and prepare students to engage with tools and techniques that may have career enhancing benefits in many ways.

REFERENCES

Chwif, L., M. R. P. Barretto, and R. J. Paul. 2001. "Assessment of Student Preparation for Discrete Event Simulation Courses." In *Proceedings of the 2001 Winter Simulation Conference*, edited by B. A. Peters et al., 1624–1631. Piscataway, New Jersey: IEEE.

Ginder, S. A., J. E. Kelly-Reid, and F. B. Mann. 2017. "Postsecondary Institutions and Cost of Attendance in 2016-17; Degrees and Other Awards Conferred, 2015-16; and 12-Month Enrollment, 2015-16: First

Look". *NCES 2017-075*. National Center for Education Statistics. Washington, D. C.: NCES.
Greenwood, A. G. 2017. "Striving for Ubiquity of Simulation in Operations Through Educational Enhancements". In *Proceedings of the 2017 Winter Simulation Conference*, edited by V. Chan et al., 4252–4263. Piscataway, New Jersey: IEEE.
Harling, J. 1958. "Simulation Techniques in Operations Research: A Review". *Operations Research* 6(3):307–319.
Jacobson, S. H., D. J. Morrice, D. H. Withers, E. Yucesan, and W. D. Kelton. 1994. "Teaching Simulation: A Panel Discussion". In *Proceedings of the 1994 Winter Simulation Conference*, edited by M. S. Manivannan et al., 1378–1381. Piscataway, New Jersey: IEEE.
Jentsch, F., and M. Curtis. 2017. *Simulation in Aviation Training*. London: Routledge.
Katsaliaki, K., and N. Mustafee. 2015. "Edutainment for Sustainable Development: A Survey of Games in the Field". *Simulation & Gaming* 46(6):647–672.
Lateef, F. 2010. "Simulation-based Learning: Just Like the Real Thing". *Journal of Emergencies, Trauma, and Shock* 3(4):348-352.
Lieberman H., F. Paternò, M. Klann, and V. Wulf. 2006. "End-User Development: An Emerging Paradigm". In *End User Development: Human-Computer Interaction Series*, vol. 9, edited by H. Lieberman, F. Paternò, and V. Wulf, 1-8. Dordrecht: Springer.
McGuire, C. 1976. "Simulation Technique in the Teaching and Testing of Problem-Solving Skills. *Journal of Research in Science Teaching* 13(2):89–100.
McHaney R. W. , L. Reiter, and I. Reychav. 2018. "Immersive Simulation in Constructivist-Based Classroom E-Learning". *International Journal on E-Learning* 17(1):39–64.
McHaney, R. W. 2009. *Understanding Computer Simulation.* Copenhagen: Ventus/Bookboon.
McHaney, R. W. 2011. "Knowledge Spaces for Online Discovery Learning". In *Constructing Self-Discovery Learning Spaces Online: Scaffolding and Decision Making Technologies,* edited by S. Hai-Jew*,* 72-94. Hersey, Pennslyvania: IGI Global.
Mørch, A. I., V. Caruso, and M. D. Hartley. 2017. "End-User Development and Learning in Second Life: The Evolving Artifacts Framework with Application". In *New Perspectives in End-User Development*, edited by F. Paternò and V. Wulf, 333-358. Cham: Springer.
Robinson, S. 2008. "Conceptual Modelling for Simulation Part I: Definition and Requirements". *Journal of the Operational Research Society* 59(3):278–290.
Robinson, S. 2014. *Simulation: The Practice of Model Development and Use,* 2nd ed. London: Palgrave.
Shin, H., H. Ma, J. Park, E. S. Ji, and D. H. Kim. 2015. "The Effect of Simulation Courseware on Critical Thinking in Undergraduate Nursing Students: Multi-Site Pre-Post Study". *Nurse Education Today* 35(4):537–542.
Stahl, I. 2000. "How Should We Teach Simulation?". In *Proceedings of the 2000 Winter Simulation Conference*, edited by P. A. Fishwick et al., 1602–1612. Piscataway, New Jersey: IEEE.
van der Zee, D. J., M. Pidd, A. Tolk, K. Kotiadis, A. A. Tako, O. Balci, and M. Elder. 2010. "Panel Discussion: Education on Conceptual Modeling for Simulation-Challenging the Art." In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson et al., 290-304. Piscataway, New Jersey: IEEE.

AUTHOR BIOGRAPHY

ROGER MCHANEY is a Professor of Management Information Systems and University Distinguished Teaching Scholar at Kansas State University, College of Business. He holds the Daniel D. Burke Chair for Exceptional Faculty. McHaney is the author of books on simulation, education technology and Web 2.0 applications. Prior to becoming an academic, he developed simulations primarily in manufacturing and materials handling. His current research includes understanding how simulation and modelling techniques can best be applied in business organizations, and data-driven simulations. His email address is mchaney@ksu.edu.