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

MODELING FOR EVERYONE: EMPHASIZING THE ROLE OF MODELING IN STEM EDUCATION

Paul Fishwick

Creative Automata Laboratory
Arts & Technology
University of Texas at Dallas
800 West Campbell Road, AT10
Richardson, TX 75080-3021, USA

Sally Brailsford

School of Management
University of Southampton
Southampton SO17 1BJ, UK

Simon J. E. Taylor

Modelling & Simulation Group
Department of Computer Science
Brunel University
Uxbridge, Middx, UB8 3PH, UK

Andreas Tolk

SimIS, Inc.
200 High Street #305
Portsmouth, VA 23704, USA

Adelinde Uhrmacher

Institute of Computer Science
University of Rostock
Albert-Einstein Street 22
18052 Rostock, GERMANY

ABSTRACT

Modeling is a creative activity known and practiced by most in industry, government, and academia. A model is a construct of language coded using different technologies—from print and physical media to computer-generated synthetic environments. The activity of modeling is so central to human cognition, that we must ask ourselves whether modeling should be more clearly emphasized in education, especially within science, technology, engineering, and mathematics (STEM) fields. Coding has recently been suggested as a skill that everyone needs to acquire for the 21st Century. Can modeling be situated along side coding? We explore the connections between modeling and coding, and stress the importance of model literacy for all.

1 INTRODUCTION

We present a panel of five seasoned modelers who regularly attend different events related to modeling and simulation. The goal for the panel is to see where we share views, and where we differ in our understanding, and approach to, modeling.

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

We have chosen the word "modeling" in the paper rather than the more specific phrase "simulation modeling" as a way to broaden the discussion of modeling that is centered on the modeling of dynamic systems for simulation, but is not limited to such systems (e.g., increasingly, modeling within our discipline involves tight integration with other types of models such as those capturing abstractions of geometry and knowledge).

Brailsford begins the discussion within a specific application context – modeling for use in healthcare. She notes that the use of modeling in healthcare seems less than in other areas where modeling has a strong foothold. She also notes that modeling requires a different sort of mindset than that in a quantitative or mathematical field, and this gives hope to those who may find mathematical notation daunting. In Section 3, Fishwick characterizes modeling as a type of language, albeit an artificial type of language rather than a natural one such as English or Chinese. He then introduces the idea that in computer science modeling might play a more central role than is generally acknowledged, and that if everyone should learn "how to code," then perhaps everyone should also learn "how to model" given that modeling is a broader activity including structural representations of knowledge, dynamics, and geometry. Taylor, in Section 4, notes that there has been a long history of attempts to introduce computing into the whole span of the curriculum in the UK. Taylor notes that the ease in which one can code is often associated with the ability to model something interesting and motivating — robot characters is one example. Therefore modeling in an applied sense (e.g., building micro-worlds) is already underway from an early age but not usually followed through into Universities. In Section 5, Tolk observes that modeling as a method is widely used and broader than we might realize. He notes that models are at the heart of science, and so our scientific progress is an evolutionary series of models. He raises the importance of an epistemology of modeling and the essence of contextualizing how and what we model within a philosophical framework. In Section 6, Uhrmacher notes differences in how fields structure knowledge. For example, in biology the structures may be descriptive, and in engineering, more formal. Past history of science suggests a gradual transition from descriptive to formal. She suggests that to assist in making modeling more ubiquitous, we should stress more domain-specific modeling languages—languages that speak to specific groups for targeted domains.

2 SALLY BRAILSFORD

In my section I focus on one specific application area, modeling in healthcare, and discuss an issue which has interested me for some time. Despite a massive academic literature in healthcare simulation modeling, and a few notable counterexamples (mainly in the US), there has been relatively little evidence of widespread adoption and implementation of simulation in practice by healthcare provider organizations (Fone et al. 2003). Compared with other sectors such as manufacturing or defense, modeling and simulation are simply not part of healthcare management decision-makers' routine toolkit (Brailsford et al., 2009). The question I shall discuss is, suppose modeling were to be a standard part of a medical education, would this help to increase adoption? Should this just be for doctors (who are required to study basic statistics already) or should we also include nurses and other professionals – pharmacists, radiologists and so on? Should we also include hospital CEOs and managers? How about analysts?

For many years I have worked predominantly in the field of healthcare simulation, in a variety of applications, ranging from more traditional patient flow applications (how many nurses does the Emergency Department need? How many beds does the Intensive Care Unit need?) to rather more unusual applications, such as modelling a disease process in an individual and then using this model to compare and evaluate different treatments or interventions, in a kind of simulated clinical trial. One of the main factors that these models have in common is that they have all involved collaboration with clinical health professionals, managers and analysts, and thus I have frequently begun projects by having to explain what a model is. My current favorite example is to show two screenshots from Google maps of the area near my house, the map view and the satellite view. I begin by saying they are both models: they are not real streets in Southampton, but are both simplified versions of the real streets with many details left out (especially the map). However they are both useful in solving different problems, depending whether they are trying to find their way to my house by car, or trying to spot my house as their plane comes in to land at Southampton airport.

One interesting thing I have observed is that most clinicians are fairly comfortable with the concept of modeling, whereas managers are less happy with it (unless they happen to have a previous background in industry) and analysts, who are in love with data, least of all. Needless to say, it is a rare clinician who has any knowledge of coding or who has come across computer simulation. However in my experience clinicians very quickly grasp the concept of modeling, i.e. developing a computer program which captures the essential features of the real system and then using the program to conduct experiments before trying something for real. The visual aspects of discrete-event simulation are very helpful here. Maybe too helpful, since it is the next step – deciding which features are essential – which is generally more problematic. Clinicians tend to want to include everything in a model and always ask for more detail. Possibly, this is because they know from experience (or have it drummed into them during their medical training) that everything does matter.

In 2010 I led a project (Brailsford et al. 2013) which undertook a qualitative survey on behalf of the UK National Health Service's Institute for Innovation and Improvement (III). The aim of the study was to evaluate the adoption and usefulness of a specific modeling tool, a variant of Simul8 (www.simul8.com) called Scenario Generator (SG) which had been provided free of charge to selected NHS organizations for one year by the III. Only one license was provided per organization, and different organizations had given the software to different people. As part of this project we interviewed a variety of NHS staff – doctors, nurses, analysts, managers and public health specialists – and in addition to discussing SG specifically, we also explored their understanding of, and previous experience with, modeling in general. One of the most interesting findings was that it was not always the most obvious person who most quickly grasped the concept of modeling and saw the potential benefits of simulation. In other words, it was not always the most technically-minded person or the person with the strongest background in IT, mathematics or statistics. Indeed those organizations where this type of person had been given the software were often those who had used the tool the least. We found that it was often the personality type of the main user, rather than technical or analytical background and skills, which determined whether or not SG had been used successfully in practice. This leads one to suspect that modeling, as distinct from coding or data analysis, requires a different mindset – the ability and vision to abstract the key aspects of a system or problem and communicate this with others. This is an encouraging message for educators of STEM subjects, since it suggests that modeling could be a useful way to attract and excite the sort of students who lack confidence and claim to "hate math" because they were never any good at arithmetic in school.

3 PAUL FISHWICK

Should everyone model? We all do early in life in the form of scale models. Toys are often in the form of scale models, however, these sorts of models are still in use for science and engineering. When we learn that two blocks can be modeled by the character "2", we are employing the technologies of print rather than those of wood making. In using these characters, we use symbolic links rather than physical similarity. The ubiquity of modeling can also be framed within the movement of learning to code (Code 2014; Ycode 2014). What is the connection between code and modeling? We'll discuss the topic of defining models and code. We then progress to a broad view of modeling that encompasses information, geometry, and dynamics.

Models are the product of languages (Fishwick 2007), allowing us to frame our cognition and communicate to each other. For example, a queuing network for capturing the dynamics of kitchen staff is a product of the language of queuing networks. The network may be represented in mathematical notation, as a diagram, or something else entirely. It would seem logical to ask ourselves whether a queuing network represented in a textual notation is a different language than the same network which is in the form of a diagram. Regardless of how we demarcate our languages, models can be represented using different media. This condition is made a bit more complex considering that, like natural languages,

languages used for modeling can have dialects. A queuing network developed by one set of researchers can support specific nodes that some other queuing networks lack. A definition of "code" depends on how broadly we define that term. Common parlance suggests that code is written in the form of a formal, written or printed, language. With that definition, coding can be construed to be modeling with a certain type of textual language rather than one that is in another representational form such as tangible, virtual, or diagrammatic.

Having framed models in terms of languages, what sorts of models are relevant within the greater modeling and simulation communities of the sort associated with these proceedings? The Winter Simulation Conference (WSC) historically promotes a wide array of modeling languages and products. The vast majority of the modeling types presented in our community are characterized as relating to dynamics or behavior (Fishwick 1995). Queuing networks model how systems containing discrete events and shared resources function. These models are useful in abstracting away from the system everything except for objects that contend for limited resources, resulting in queuing behaviors. These networks are defined as mostly static structures for dynamic operations. It is also possible to use other types of models for capturing these behaviors. For instance, agent-based modeling is a way to encode dynamics, not as a network, but as a set of parallel entities with each entity having its own behavior. The entity, or agent, approach is especially useful in situations where space is explicit. In a queuing network, space is abstracted away. In an agent-based system, agents must be aware of the geometric conditions in which they navigate. Although models of behavior, like those of queuing networks and agent-based systems, dominate the culture of WSC, there are other modeling categories that are important to consider.

Two additional types of models that are in wide use within WSC are models based on information and geometry. Information models can be represented conceptually as a list of propositions and predicates, or as a semantic network. Often these information models take on the aura of models of knowledge especially if there are mechanisms to infer new knowledge. The queuing network that defines task flow within a kitchen may start out in a more general way looking like a semantic network which contains classes, instances, and relationships. The queuing network can be seen as a byproduct of this network where we select specific types of relations and a subset of kitchen objects. Geometric models are also useful if we are to visualize a simulation. Denoting a queue can be geometrically modeled using dots, lines, or 3D rendered objects that resemble the queued items being modeled.

The concept of computer science being a model-based empirical science has been raised (Fishwick 2014), and leads to a greater need for everyone to model, thus augmenting the thrust to educate people how to code. If code is modeling with text, and if computing is around us waiting to be observed in the wild, then the need to model is a strong one. We should be looking for ways to teach modeling in K-12 and in universities. This panoramic view of modeling goes beyond seeing modeling and simulation at the periphery or only as an application domain.

4 SIMON J E TAYLOR

What can M&S contribute to education? I argue that it should be part of educational programs from a very early age, at least in the way we frame how we interact with computers and experience learning. There is a widespread availability of internet connected devices. In some cases, infants and young children are using these devices, especially touchscreen tablets and smartphones, on a regular basis (Holloway, Green and Livingstone 2013). While security and digital footprints are a concern for all parents, and access to these devices is by no means uniform across society, what can the regular use of ICT by children from a young age mean for M&S?

Developing software and hardware to help children to learn to use and program computers is not a new concepts (see for example the BBC Micro (http://www.bbc.com/news/technology-15969065) and Sinclair's computers(http://oldcomputers.net/zx80.html)). However, the Web has made possible the development of a new range of simplified languages to introduce people to programming. One great example is Scratch (scratch.mit.edu) (Maloney, et al. 2010). This is a free programming language developed by the Lifelong Kindergarten Group at MIT that helps children to program interactive media such as stories, games and animation. Scratch programs are developed by dragging and dropping "blocks" from a predefined palette and attaching them to other blocks like a jigsaw puzzle. Multiple block structures are called "scripts". Users program in Scratch by dragging blocks from a block palette and attaching them to other blocks like a jigsaw puzzle. Structures of multiple blocks are called scripts. This method of programming (building code with blocks) is referred to as "drag-and-drop programming". Scratch programs can interact in a sandbox with keyboard and mouse, ambient volume and video sensing (movements provided as sensor values from a webcam). The Scratch environment gives plenty of visual feedback when running and debugging a program. The language is aimed at eight to sixteen year olds.

Building on experiences from Scratch, ScratchJr (www.scratchjr.org) is an even more introductory visual programming language aimed at younger children (ages 5-7). The focus is on the development of interactive stories and games with a focus on problem-solving skills as well as early stage literacy and numeracy. A simplified block language is used to make characters move, jump, dance and even sing.

Other devices exist that make the connection to the physical world. The "Finch", for example, is a low-cost little robot developed by Carnegie Mellon University for computer science education (www.finchrobot.com). It was designed to help students to engage with programming. It has support for over a dozen programming languages and environments (including Scratch and other programming environments aimed at school children). It has light, temperature and obstacle sensors, accelerometers, motors, a buzzer and a full color beak LED. Working with local schools, the Department of Computer Science at Brunel University launched the Adopt-A-Bot competition (www.adoptabot.org.uk). School teams used the Finches to tell a story and programmed them to act as "babysitters", repair crew on space craft working in a vacuum, nuclear power station engineers and as night watchmen able to raise the alarm in burglaries and to spot the earliest sign of fire. Other programmable robotics and sensor systems exist such as Lego Mindstorms (mindstorms.lego.com) and the Raspberry Pi (www.rasberrypi.org). The Leap Motion (www.leapmotion.com) is a sensor that allows gesture control and manipulation of programs, and is another example of the many low cost gizmos that are available.

With more and more of these free dedicated programming environments, (relatively) low cost robotics, computers and sensors, and the widespread availability of internet enabled devices, children today have remarkable exciting opportunities to create and simulate models of their world. Children still play board games (especially in Germany) but also have range of computer games to choose from. Computer games and consoles are interacting more and more with the physical environment. Indeed some hybrid games are emerging that combine elements of board games and computer games and others are appearing that use a Scratch-like approach to interact with game characters.

Skipping forward to the final year of an undergraduate degree, in many countries the final year requires students to undertake a major project. In my experience some of the most interesting and adventurous have been projects in M&S. Two examples. Working in a team but with individual projects, seven BSc (HONS) Business Computing students worked with stakeholders at Hillingdon Hospital to study a range of clinical service problems. Each produced a range of M&S artefacts that compared AS-IS and TO-BE system designs and included storyboards, business process models, simulation models and decision support tools. Each student made a formal presentation to the hospital at the end and the best and runner up projects were awarded. Similarly, four Computer Science students took part in the international Simulation Exploration Experience (SEE) (exploresimulation.com), supported by NASA and a range of simulation vendors. Around ten other Universities took part. The goal of SEE is to promote distributed simulation and the IEEE 1516 High Level Architecture in STEM education and provides a Lunar environment for students to develop their simulations. The students developed a hybrid distributed simulation of a Lunar Mining Operation consisting of a discrete-event simulation (factory), real-time simulation (astronaut) and a mine (agent-based simulation). Other Universities developed rover, spacecraft, space stations and an asteroid shield. The students were encouraged to collaborate and to conduct themselves in a professional team-basis. SEE came together at an international event that ran in parallel with SpringSim 2014 in Tampa. NASA provided a visualization tool based on Unity that showed the simulations working together in a Lunar (and near-Lunar) environment. Students involved in these projects all agreed that they gained many useful skills and experience from these projects (as did the stakeholders). I am sure that these experiences are repeated in student M&S projects around the world in many STEM subjects.

From an educational perspective, in the UK at least, there is a move to shift from ICT education to Computer Science. It aims to teach children as young as five how to create and debug simple programs and then for children between seven and eleven children to design, write and debug programs that accomplish specific goals. The Adopt-A-Bot competition is an example that illustrates how excited children were when they took part in using some of the technologies outlined earlier to model and simulate a story. Importantly, children at school are using more and more sophisticated technologies. There is a huge opportunity for M&S here to drive school-level Computer Science education through problem-based learning. The examples of final year projects serve to show how useful M&S problem-based learning can be to students and, more widely, to STEM education. How can we match the expectations of school leavers in the future with STEM curricula that have elements of M&S problem-based learning, as well as the more formal theoretical skills needed to support University-level M&S education? How can we engage educators across schools and Universities? What is needed urgently is widely available educational material (and tools) that supports M&S-led education across all ages ranging from, for example, Scratch for M&S to examples of successful M&S final year projects combined with efforts from Universities to reach out to school communities.

5 ANDREAS TOLK

Our use of models is already much deeper rooted than many people think. From early school days, students learn how to apply the scientific method: In order to understand a phenomenon we formulate a hypothesis that predicts the outcome of an experiment. If the outcome is actually observed, the hypothesis is supported and becomes an explanation and ultimately can contribute to a theory. In the process of defining the experiment, we select to observe only those parts of reality that we believe to be relevant regarding our hypothesis. In addition to this simplification, some parameters are not observed with the full accuracy or detail possible, but we build aggregates that we take into account to focus on what is important for our hypothesis. By our prediction, we connect input parameters functionally with the independent parameters of our experiment, establishing causality. In other words: we are building a model, a purposeful simplification and abstraction of reality. The scientific method, theory building, and theory testing are going hand in hand. As a scientist, you are a modeler. Every scientific theory is a model. Models are the heart of science. The history of science is a series of models (Goldman 2006).

This insight is not new, and science philosophy is dealing with modeling and the use of models to capture knowledge as well as to gain knowledge. Karl Popper (1935) introduced the three worlds to explain why we need models: the *physical world of objects*, the *mental world of conceptualizations*, and the *formal world of objective knowledge*. When we are addressing the real world, we are using our conceptualization thereof to make the arguments. In order to get insights, we build a formal model to capture and code our knowledge. In the same era, Ogden and Richards (1923) introduced the semiotic triangle that distinguishes between the real world referent that is observed, the conceptualization we have about the real world referent, and the symbols we use to communicate our conceptualizations. Again, the model is the cognitive part, the result of our conceptualization of our observation. In his essays on life itself, Robert Rosen (1998) states: *"I have been, and remain, entirely committed to the idea that modeling is the essence of science and the habitat of all epistemology."* In summary, models are well-known to be useful for ontological and epistemological purposes, and recent Winter Simulation Conference activities refocused on these epistemological roots (Tolk et al. 2013).

If we agree that models are the result of purposeful simplification and abstraction of the truth resulting in a conceptualization that replaces observable correlation with causality, the obvious next question is: what shapes our ability to model? Even if we take a positivistic worldview and assume that only one observable reality exists, and post-positivism and relativism are challenging this premise, our perception is shaped by physical-cognitive aspects and constraints. The *physical* aspect defines what attributes of an object are observable with the sensoric system of the observer, or more general, the information about the object that can be obtained in the process of perception (this can include gaining insight from literature, discussions with colleagues, using instruments, etc.) The *cognitive* aspect is shaped by the education and the knowledge of the observer. In order to conceptualize the observation the observer needs to have an internal model. The subject matter expert of a domain has more internal models to explain an observation in his field than other persons. Even if two modelers are building models of exactly the same observation, they are more than likely coming up with two models.

The question posit in the introduction of this position paper goes deeper, as a model needs to be expressed in a language. The question is plausible to ask if a model expressed two different paradigms is still the same model? The product of our conceptualization processes is often referred to as the conceptual model. Robinson defines that "*a conceptual model is a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications of the model"* (Robinson 2007, p. 283). He argues that the conceptual model establishes a common viewpoint that is essential to developing the overall model. But to what degree is the model shaped by the paradigm the modeler is comfortable with? When he/she observes traffic in a busy city street, is he/she seeing an agent population of individual cars or a wave of objects moving through the streets? Will or even should a system dynamics modeling expert start with the same model as a discrete event simulationist or and agent-based simulation expert? Which paradigm is better? Which one should we teach?

Another aspect is added when we are building computer simulations: they are limited to problems that can be decided by an algorithms using computable functions, and these problems are only a small subset of what we like to understand. If we are interested in simulations that end in a reasonable amount of time, we are cutting our set further back. Finally, we are interested in consistency. The same logical fact should not result in different answers based on where it is evaluated at a given time. The consistent representation of truth in all participating systems of a federation – or also components of a system – was therefore recommended as the definition for composability to complete the definition for interoperability, which only requires that data can be exchanged between systems or components and can be used on the receiving side (Tolk 2013). Inconsistent theories can therefore not belong to the same model that results in a simulation. If we have more than one possibility theory, we will need more than simulation to evaluate a challenge, as every simulation can best be understood as an "executable theory."

Is modeling and coding for everyone possible or even desirable under these constraints? I do think so. We may not be able to continue to be so naïve that we try to sell M&S as the general solution to everything, or even a new way to gain insight in science. Instead, we have to emphasis that each model can always only capture one facet of a problem to be solved, and a well-orchestrated set of tools will be necessary to address non-trivial problems systematically, systemically, and holistically. In order to reach this objective, modeling needs to be emphasized as the central creative part in our curricula. The earlier modeling, but also simulation, is integrated into the education process, the better students will understand and be able to utilize these tools to preserve and communicate knowledge. A positive step is the high school level curriculum developed under the leadership of NASA (Raiszadeh and Batterson 2012).

6 ADELINDE UHRMACHER

6.1 Should everybody (or nearly everybody) do modeling?

In his work "logic of failure" (Dörner 1997), Dietrich Dörner delineates pitfalls of strategic thinking, people stumble into when faced with a dynamic system. In general non-linear dynamics appear to be a problem for the human mind. Similarly as people tend to be cognitively biased when dealing with probabilities (Tversky and Kahneman 1974), they only get it seldom right when confronted with non-linear dynamics. So for Dietrich Dörner modeling and simulation is an indispensable tool to reveal these shortcomings in human strategic thinking and, consequently, to do something about it.

In 2002, Yuri Lazebnik asked “Can a Biologist Fix a Radio?” (Lazebnik 2002) to summarize that by the experimentalistic approach that biologists take it would be rather unlikely to understand the working of a radio – which is rather simple in comparison to cell biological systems – yet engineers would be able to do so. As reason of the failure respectively success, Yuri Lazebnik identifies the language both disciplines are using. Whereas Biologists take a rather descriptive approach, engineers use a formal language. For Yuri Lazebnik and for many other researchers in Biology, complementing experimental wet-lab studies by formal modeling and simulation, appears crucial to move the field ahead. So in both cases above, the answer to the question ”‘should nearly everyone model”’ would be a definite yes – but how can we make this happen?

6.2 How to make it happen - domain specific modeling languages

As Wittgenstein stated “the limits of my language mean the limits of my world,” one might be tempted to advocate a rather general modeling approach. However, modeling in a general approach is rather cumbersome and they hardly qualify for “a simple language that ... scientists can use to introduce themselves to formal descriptions of ... processes” and whose role it would be to help in “overcoming a fear of long-forgotten mathematical symbols.” (Lazebnik 2002). To develop such a simple language, syntax as well as semantics matter. E.g. in (Henzinger, Jobstmann, and Wolf 2011), different formalism for specifying Continuous Time Markov Chains are analyzed referring to compositionality, expressiveness, succinctness, and ease of use. All of the formalisms, e.g., stochastic Petri Nets and stochastic process algebras, describe Continuous Time Markov Chains. Thus, they only vary with respect to syntax-related aspects. However, the “how something can be expressed” largely determines “what will be expressed.” We also find rule-based languages whose syntax appears quite similar, but which are mapped to boolean, deterministic, stochastic, or even Brownian movement semantics. The semantics will determine which and how questions can be answered. Thus, to support a modeling for (nearly) everyone, domain-specific languages are asked for, that are tailored specifically to an application domain offering notations, abstractions, and an expressive power focused on a particular problem domain (van Deursen, et al. 2000).

However, even if those domain-specific modeling languages are in place, are we yet there? One decade after the paper by Yuri Lazebnik, James Faeder states in his commentary to a rule-based multi-level modeling language (Maus, Rybacki, and Uhrmacher, 2011) “the development of rule-based modeling languages and tools, ... , in recent years represents a near-fulfillment of Lazebnik’s vision of precise formal modeling languages for biology” (Faeder 2011). However, he admits that despite these developments, still only few experimental biologists would be familiar with modeling and even less would be actively involved. Having a simple to use and yet expressive modeling language appears only to be part of the answer. To develop a model, experiments are needed, which again require, in addition to efficient and effective methods, knowledge about methods to be used and the general process of these “in-silico” experiments. Thus, in addition to domain-specific modeling languages suitable intelligent support in experimenting is required as essential ingredient to realize modeling for (nearly) everyone.

7 SUMMARY

It has been suggested that coding is a key skill for the 21st Century. If this is the case then one might ask the question, “What are we coding for?” M&S can provide fascinating and motivating problems that require a range of skills to solve. The ability to code allows us to express ourselves in a language that helps to engage with problem-solving from an early age and to begin to develop key analytical skills that would be broadly useful across society. M&S may therefore been seen as a motivating reason to code and coding as an bridge to developing further skills needed in STEM education and beyond. This paper has brought together a diverse range of views on the impact of M&S in STEM education that we hope will continue to stimulate this debate.

We have raised a number of questions, which seems appropriate for a panel discussion with a philosophical topic of modeling and its role in education. The following represents a partial list of questions raised by the panelists:

- Should we grow and nurture our philosophy of modeling and simulation?
- Is Computer Science a model and/or code-based discipline?
- Should models be emphasized in concert with the current emphasis "to code?"
- What information do we include and leave out of models for education?
- What sorts of people are most easily educated in practice of modeling?
- Should modeling as a discipline, or area of study, be required in K-16 education?
- Should models be customized by domain of use?
- Should there be standards for models used in education?

REFERENCES

Brailsford SC, Harper PR, Patel B and Pitt M (2009). "An Analysis of the Academic Literature on Simulation and Modelling in Healthcare." *Journal of Simulation* 3:130-140.

Brailsford SC, Bolt TB, Bucci G et al. (2013). "Overcoming the barriers: a qualitative study of simulation adoption in Primary Care Trusts in the NHS." *Journal of the Operational Research Society* 64:157-168.

Code. 2014. "Code" http://code.org. Last accessed May 4, 2014.

Doerner, D. 1997. *The Logic of Failure: Recognizing and Avoiding Error in Complex Situations*. Basic Books.

Faeder, J. 2011. "Toward a comprehensive language for biological systems". *BMC Biology* 9 (69).

Fishwick, P. A. 1995. *Simulation Model Design and Execution: Building Digital Worlds*. Englewood-Cliffs, New Jersey, Prentice Hall, Inc.

Fishwick, P. A. 2007. "The Languages of Dynamic System Modeling, Chapter I." In *Handbook of Dynamic System Modeling*, Fishwick, P. A., Ed., Chapman and Hall/CRC Computer & Information Science Series.

Fishwick, P. A. 2014. "Computing as Model-Based Empirical Science." In *Proceedings of the 2014 SIGSIM-PADS Conference*, Hamilton, D., Riley, G. and Fujimoto, R., Eds. Association for Computing Machinery, Denver, Colorado, May 18-21, 2014.

Fone D., Hollinghurst S., Temple M., Round A., Lester N., Weightman A., Roberts K., Coyle E., Bevan G. and Palmer S. (2003). "Systematic review of the use and value of computer simulation modelling in population health and health care delivery." *Journal of Public Health Medicine*, 25:325-335.

Goldman, S.L. 2006. *Science Wars: What Scientists know and How They know It.* Lehigh University, The Teaching Company, Chantilly, VA

Holloway,D., Green, L. and Livingstone, S. (2013). *Zero to eight. Young children and their internet use*. LSE, London: EU Kids Online.

Koksal, A. S., Y. Pu, S. Srivastava, R. Bodik, J. Fisher, and N. Piterman. 2013. "Synthesis of Biological Models from Mutation Experiments". *SIGPLAN Not.* 48 (1): 469–482.

Lazebnik, Y. 2002. "Can a biologist fix a radio?–Or, what I learned while studying apoptosis.". *Cancer Cell.* 2 (3): 179–182.

Maloney, J., Resnick, M., Rusk, N., Silverman, B. and Eastmond, E. 2010. The Scratch Programming Language and Environment. *ACM Transactions on Computing Education (TOCE).* 10 (4): Article No. 16.

Maus C., Rybacki S. and Uhrmacher A.M., 2011. Rule-based multi-level modeling of cell biological systems. BMC Systems Biology, 5:166.

Ogden, C. K., and I. A. Richards. 1923. *The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism*. London: Routledge & Kegan Paul.

Popper, K. R. 1935. *Logik der Forschung* [*The Logic of Scientific Discovery*], Berlin: Springer.

Raiszadeh, B., and J. Batterson. 2012. "CK-12 Modeling and Simulation for High School Teachers: Principles, Problems, and Lesson Plans." http://www.ck12.org/book/CK-12-Modeling-and-Simulation-for-High-School-Teachers%253A-Principles%252C-Problems%252C-and-Lesson-Plans/

Rosen, R. 1998. *Essays on Life Itself.* New York NY: Columbia University Press.

Tolk, A. 2013. "Interoperability, Composability, and their Implications for Distributed Simulation – Towards Mathematical Foundations of Simulation Interoperability." In *IEEE/ACM Proceedings Distributed Simulation and Real Time Applications (DS-RT 2013)*, 3-9, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Tolk, A., B.L. Heath, M.Ihrig, J.J. Padilla, E.H. Page, E.D. Suarez, C. Szabo, P. Weirich, and L. Yilmaz. 2013. "Epistemology of Modeling and Simulation." In *Proceedings of the 2013 Winter Simulation Conference,* edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 1152-1166. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Tversky, A., and D. Kahneman. 1974. "Judgment under Uncertainty: Heuristics and Biases". *Science* 185 (4157): 1124–1131.

YCode. 2014. "YCode." http://yearofcode.org.

AUTHOR BIOGRAPHIES

SALLY C. BRAILSFORD is Professor of Management Science at the University of Southampton, UK. She received a BSc in Mathematics from the University of London, and MSc and PhD in Operational Research from the University of Southampton. Her research interests include simulation modeling methodologies, system dynamics, health service research and disease modeling, and the modeling of human behavior in healthcare systems. She is chair of the European Working Group on OR Applied to Health Services (ORAHS) and is an Editor-in-Chief of the journal Health Systems. She is on the editorial boards of Health Care Management Science, the Journal of Simulation, and Operations Research for Health Care. Her email address is s.c.brailsford@soton.ac.uk

PAUL A. FISHWICK is Distinguished University Chair of Arts & Technology and Professor of Computer Science at the University of Texas at Dallas. Fishwick was General Chair of the Winter Simulation Conference in 2000, currently serves as Chair of ACM SIGSIM, and has served on all major editorial boards in modeling and simulation including the IEEE Transactions on Systems, Man, and Cybernetics, ACM Transactions on Modeling and Simulation, Journal of Simulation, and Transactions of the Society for Modeling and Simulation International. His email address is Paul.Fishwick@utdallas.edu .

SIMON J E TAYLOR the Editor-in-Chief of the Journal of Simulation and Series Editor of Palgrave Macmillan's OR Essentials. He is a Reader in the Department of Computer Science, Brunel University, and leads the Modelling & Simulation Group. He is a former ACM SIGSIM chair and a member of the Steering Committee. His recent work has focused on the development of standards for distributed simulation and grid- and cloud-based simulation (www.cloudsme.eu) as well as the spread of the international Grid Infrastructure into Africa (www.eI4Africa.eu). His email address is simon.taylor@brunel.ac.uk and his web address is www.brunel.ac.uk/~csstsjt.

ANDREAS TOLK is Chief Scientist at SimIS Inc. in Portsmouth, VA, and adjunct Professor of Engineering Management and Systems Engineering as well as of Modeling, Simulation, and Visualization Engineering at Old Dominion University in Norfolk Virginia USA. He holds a M.S. and Ph.D. in Computer Science from the University of the Federal Armed Forces in Munich, Germany. His email address is andreas.tolk@simisinc.com.

ADELINDE M. UHRMACHER is head of the modeling and simulation group at the University of Rostock. She holds a M.S and Ph.D in Computer Science from the University of Koblenz. She is editor in chief of the ACM Transactions of Modeling and Computer Simulation. Her recent research focuses on

the development of modeling methods and support in executing experiments with applications in cell biology and demography. Her email address is adelinde.uhrmacher@uni-rostock.de.