Proceedings of the 2011 Winter Simulation Conference
S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.

ACTIVITY INTERACTION AND SELF-SIMULATING SYSTEMS

Lee Schruben
University of California, Berkeley
Berkeley, CA, 94720
and
(Honorary) Chief Science Officer
The Bioproduction Group (Bio-G)

ABSTRACT

Two radically new discrete-event simulation modeling technologies are introduced: Activity Interaction (AI), a general approach for modeling system dynamics, and Self-Simulating Systems (S^3), where AI models are integrated with information systems to generate correct, current, and credible real-time simulations. The research and development paths taken with these new technologies constitute a meta-experiment on the two fundamental historical approaches to developing new simulation modeling methodology: academic and commercial. Examples in production and service system settings are presented.

1 INTRODUCTION: TWO NEW METHODS FOR SIMULATION MODELING

This paper introduces two new methods for discrete-event simulation modeling: Activity Interaction (AI) and self-simulating systems (S^3). Examples of their application in production and service systems are presented.

This paper also contrasts the two fundamental ways new simulation modeling methodologies have historically been developed: commercial software where research and development are driven by the market, and academic research where peer-review is the driving force. Both these research approaches are supported by different branches of the National Science Foundation, and by other research institutions including the Center for Information Technology Research in the Interest of Society (CITRIS), but with very dissimilar missions.

The Activity Interaction modeling methodology was created using the non-academic approach of implementing the ideas in commercial software and developing them in the marketplace. Basic research on AI is supported in part by the Small Business Innovative Research (SBIR) program of the National Science Foundation (phases A and B), but has largely been self-supporting. While the AI software implementation is completely general, the focus has been exclusively on real problems in the critically important and highly complex processes and systems in biopharmaceutical manufacturing and supply chains where the life-saving therapies based on biologics are discovered, produced, and delivered to patients. The firm that implemented AI, The Bioproduction Group Inc (Bio-G.com), has been profitable from the beginning and has grown organically (without venture capital). Bio-G has among its customers the world's top biopharmaceutical firms and has saved its clients hundreds of millions of dollars (and in turn, saving lives), while outperforming all competitors in every confrontation in the marketplace. Bio-G's products include Crosswalk for general information system interoperability, and the Real-Time Modeling System (RTMS a.k.a. "Artemis") implementing the AI modeling methodology.

Research on self-simulating systems (S^3) has followed the usual academic route of initial testing, writing proposals, and after peer-review, successfully being supported by CITRIS. CITRIS is funding the basic research on the fundamental theory of S^3 along with controlled testing on simple examples. Current

978-1-4577-2109-0/11/$26.00 ©2011 IEEE

proof-of-concept implementation is underway for integrating AI into the state-of-the-art real-time information system at the new Emergence Department and the University of California, Davis, with Dr. Aaron Bair as co-principal investigator. An integrated real-time modeling system would vastly improve the utility of the ED information system, providing correct, current, and credible simulation for improving operations as well as evaluating policy or facility changes and disaster response strategies.

2 ACTIVITY INTERACTION

Activity Interaction (AI) is a new methodology for developing simulation models based on activity objects. A novel aspect of AI is that the elements of the system being modeled *do not need to be connected* by a simulation modeler. This contrasts with current simulation methodologies that require modelers to create connected entity processing flow paths (commonly used in commercial simulation software) or connected networks or di-graphs like Petri nets or Event Graphs (Seila et. al. 2003). The AI modeling methodology allows very large-scale, complex systems to be modeled where the number of system element connections is beyond the ability of a single person or even an experienced simulation modeling team to consider.

The most compelling reason for the commercial proliferation of the entity process flow methodology (sometimes referred to as a process interaction world view) is that it maps directly into animations of straightforward, small-scale, queueing-type systems. However, increased software animation capabilities come invariably at the expense of analytical power, limiting the simulations to "what if" descriptive runs and limiting its utility for prescriptive analysis. Major shortcomings of the process flow approach include that it can be hard or impossible to model some common real-world occurrences, like resource contention and concurrence, usage-based failures, jockeying to shorter queues, or time-bound processing sequences. More importantly, this approach does not scale efficiently to model highly-congested systems (arguable the very systems where simulations are most useful) or to highly state-dependent systems that indirectly share many common elements. Petri Nets are easy to create and understand (see Mueller et. al. (2007) for a novel application) and can draw on a large number of analytical methods, but do not scale to huge systems, and cannot easily model many of the real-world phenomena just mentioned. Event Graphs, while completely general (they have been proven to be Turing Complete) and scale efficiently to model huge systems, are abstract, and can become unwieldy in the hands of inexperienced or unskilled modelers. Since most complex real-world systems where simulation is of critical value are highly inter-dependent and can become congested, these conventional approaches are all limited in applicability.

Traditional discrete-event simulation of a dynamic system entails executing events in time order, beginning at some initial simulated time and progressing forward in time. In AI, a basic activity object is used to model system element concurrency and contention. Time sequencing is controlled by dynamic state/time-dependent activity interaction matrices. An activity object is instantiated when system elements interact concurrently. There are conditions for an activity to commence, continue, and terminate that are associated with state changes and time intervals in the system elements.

Since in real systems, many activities typically occur in parallel, time advance with the AI approach can be done very efficiently. Using this approach, RTMS can simulate an entire year's operations of a biopharmaceutical production facility at almost any level of detail in approximately 1 second on a laptop computer (this competes with other simulation models that run orders of magnitude slower). Such things as indirect labor (supervision and accounting) can be included easily without altering other activities, such as production or equipment changes. Indirect labor is one of the critical cost components of production systems that no conventional simulation software has previously modeled very well; an example illustrating why indirect labor is hard to model in the conventional manner is presented later in this paper.

The sequence of screen shots on the next few pages illustrates how different layers of detail are automatically simulated using AI. The system modeled here is part of a biopharmaceutical production and supply chain system. Only the model in Figure 1 is created using activity objects – the details in the next two figures are automatically generated by the software, allowing users to drill down to any level of detail without explicitly modeling it.

The most important feature of the AI model in Figure 1 that most of the “blocks” – here representing activity objects - are *not connected* to other activity objects. Their interactions are indirect (dependent on system state/time conditions). In fact, none of these blocks here need to be connected! The arcs shown in Figure 1 are included only to illustrate product flow (or could be used to represent resource cycles, etc.) and do not control the execution of the simulation model.

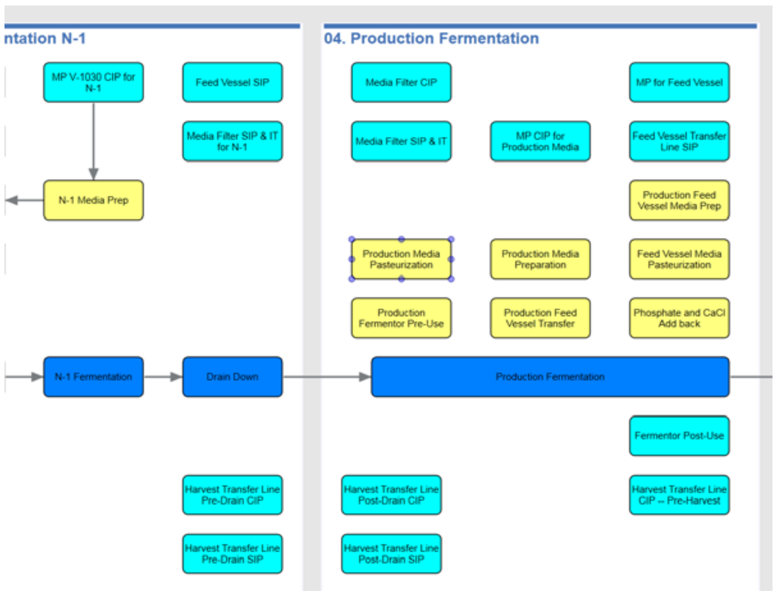

Figure 1: Top level Activity Interactions representing part of a biopharmaceutical factory.

The next two figures drill down into the detailed activity interactions the Production Media Pasteurization (PMP) activity object highlighted in Figure 1. In Figure 2 the conditions that might instantiate a PMP activity object or may be triggered by it are shown.

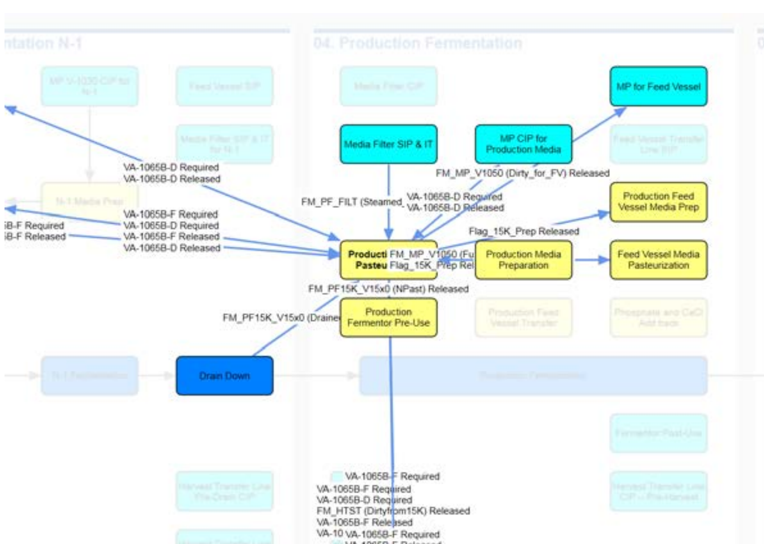

Figure 2: Conditions that may instantiate a PMP Activity object in Fig. 1.

In Figure 3, the system elements where state changes might occur during a PMP activity are shown.

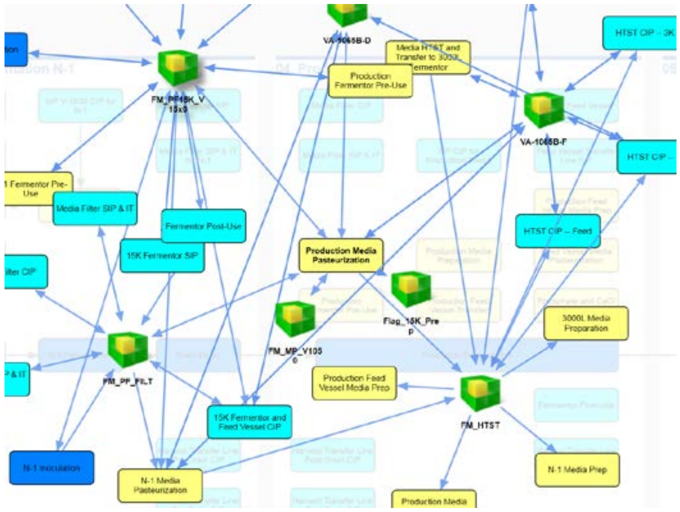

Figure 3: System elements whose activities might interact during the PMP activity in Fig 1.

The key point illustrated by Figures 1, 2 and 3 is that only the high-level activity objects in Figure 1 need to be defined; the activity interactions shown in Figures 2 and 3 are generated automatically by the RTMS software.

Figure 4 illustrates the activities of a different element in the same biopharmaceutical production system – the people. Figure 5 shows the activities that an employee (here a supervisor) might initiate, continue, or terminate. Figures 4 and 5 are both generated automatically from high-level activity objects. The point (dramatically) illustrated here is that modeling indirect labor at this level of detail using conventional modeling methodology would have required that all of the arcs in Figure 5 be specified by the modeler, in RTMS this is automatic. Clearly the complexity of modeling indirect labor at the level of detail in Figure 5 in using conventional simulation methods would be nearly impossible to do correctly. With RTMS using the AI methodology, it is straightforward and automatic. Only the high-level activity objects like in Figure 1 need to be defined. Using RTMS, it is nearly impossible *not* to do this correctly.

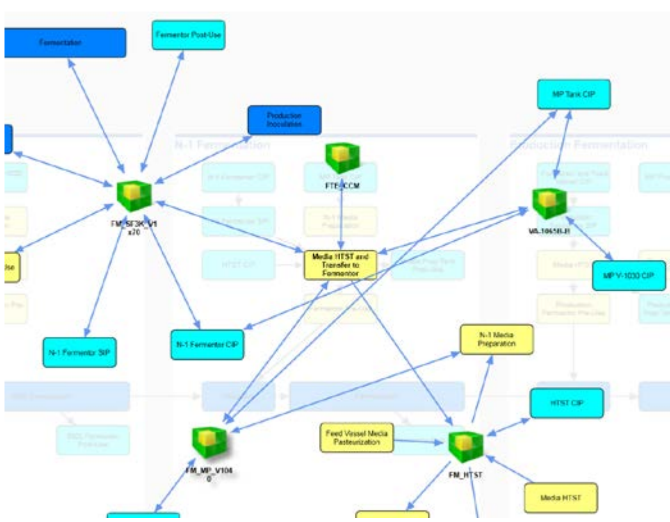

Figure 4: System elements that interact with an activity object.

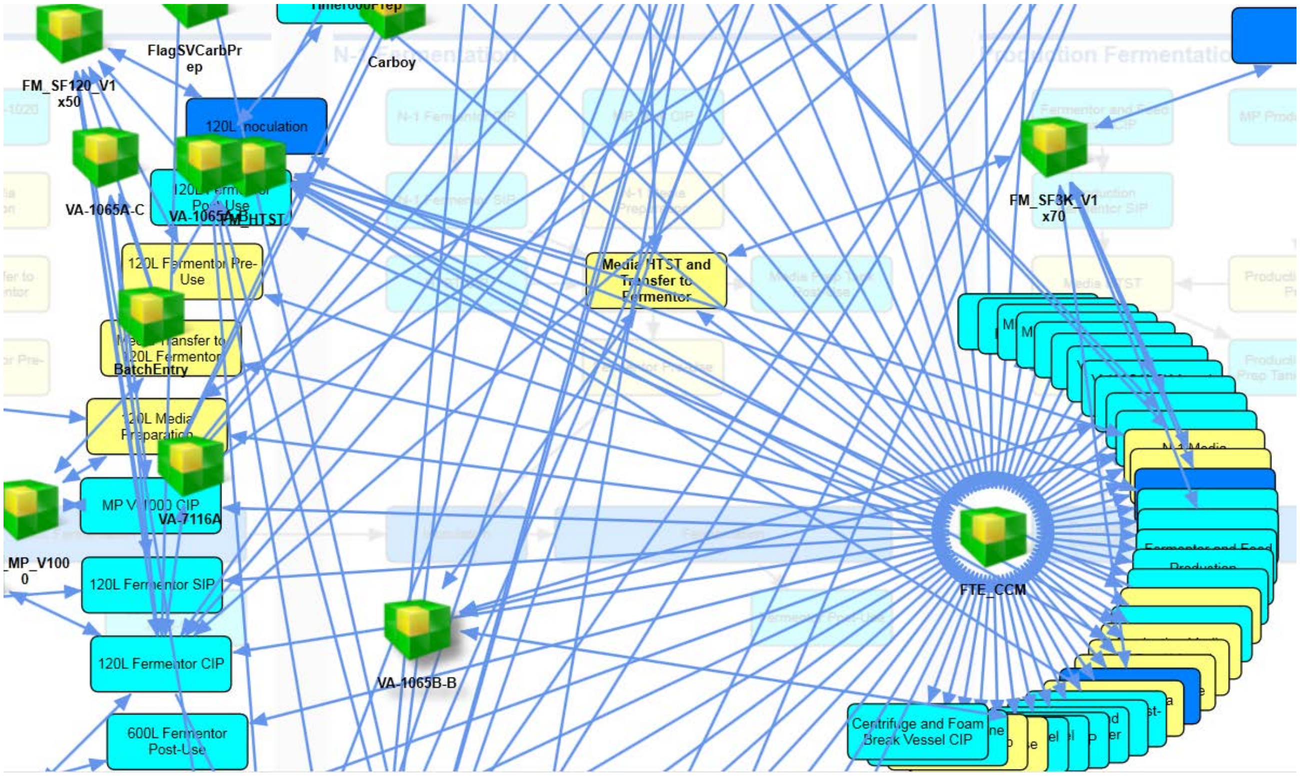

Figure 5: Activities that may be instantiated by FTE_CCM (here a person) activities

Since activity objects can be specified by anyone involved in the system it does not necessitate an outside team of simulation experts to develop the simulation model. This leads to the next, more ambitious part of this research. Designing a self-simulating system where a real-time information system automatically creates activity objects, and thus its own simulation. The terminology "information system" has caused an unfortunate confusion between data and information, they are not the same. Most real-time information systems are in fact a set of data collection systems. An integrated self-simulating information system would become truly informative by including a correct, current and credible simulation as part of its design. An ongoing research project on this topic is discussed next (results of which will be reported at the WSC 2011 conference, six months from the writing of this paper.)

3 SELF-SIMULATING SYSTEMS

Using the AI approach, it is conceptually possible for systems to simulate themselves. This should not be confused with "automatic simulation generation" or "simulation by questionnaire" proposals (Mueller et. al. 2007) or with common special-purpose simulators. (Miller et. al. 2004) A self-simulating system does what these do, but much more – it is integrated into the IT system design and adapts itself to system changes. A self-simulating system would have the advantages of being correct, current and credible since an outside team of simulation experts did not impose their system view on the domain experts. The modeler and the modeled are the same.

The system where this concept is being developed is the UC Davis Medical School Emergency Department. This is one of the largest and the most modern ED training facility in the world with the only accredited Virtual ED training programs. The system has an extensive state-of-the-art, real-time automated data systems where all resources are RFID tagged (including all equipment and personnel). Communication between staff through the numerous ED pods is instantaneous. A screen shot of one of the ubiquitous status boards in this system is shown in Figure 6.

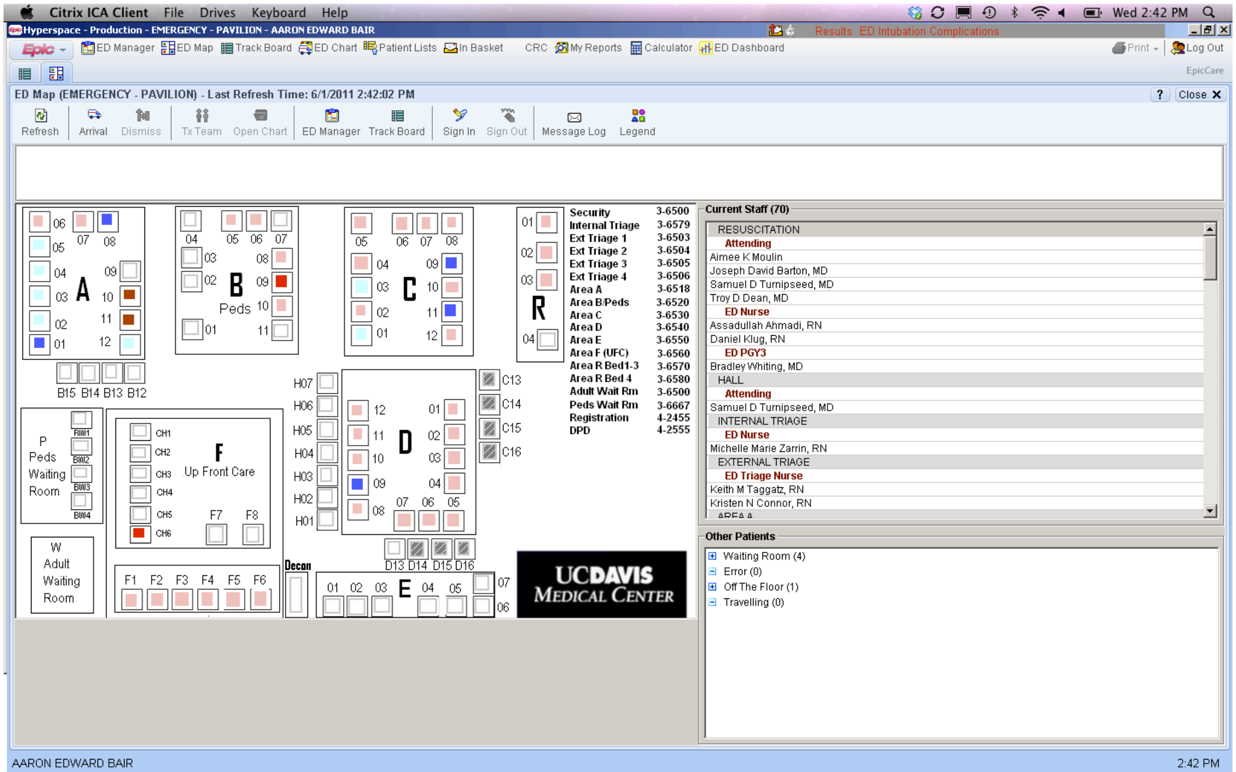

Figure 6: Status board of the UC Davis ED department

The data system can be drilled down to any level of detail at any time by authorized personnel including automatic patient medical records. Any piece of equipment can be instantly located and its activity monitored. Two-way or multiple radio communications between and among staff is instantaneous throughout the hospital.

Self-simulation involves the conceptually straightforward task of identifying activity objects, multiple system elements (staff, equipment, patients, etc) interacting concurrently and the state changes of the simulation elements involved. This involves detecting *and confirming* concurrencies of system elements in the information system.

For example: when an ultrasonograph machine, a sonographer, and a patient are at the same place and the ultrasonograph is recording an scan, then the activity of taking a ultrasound scan is occurring. This is confirmed by the earlier physician order and the subsequent scan results appearing in the patient's electronic medical record. Since each activity object is created independently (and linked by the RTMS software), the system can add or modify activity objects at any time without corrupting the activity objects already included in the simulation. The posterior distributions of the activity durations, conditioned on their continuation and termination conditions can be updated as these activities occur. It is unrealistic to expect that a perfect self-simulating system can be created this first try. It is the objective of this research project to create such a simulation with minimal interaction with the actual system and determine how information systems can be re-designed to make S^3 an effective and efficient reality.

4 A CONVENTIONAL EMERGENCY DEPARTMENT SIMULATOR

A simulation of the UC Davis emergency department was created in the conventional manner by an undergraduate team using Sigma (free for education and research at sigmawiki.com). This model is to be used for partial verification of the self-simulation. The conventional simulation is shown in Figure 7.

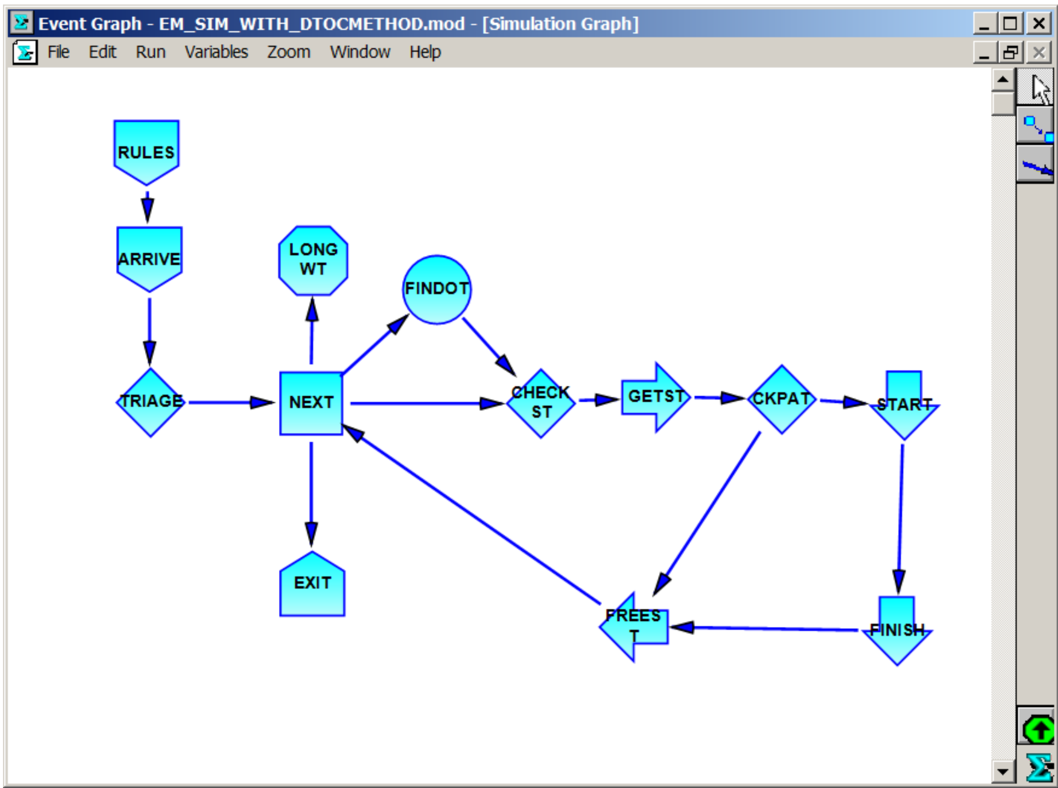

Figure 7: Data-driven general ED model in SIGMA

The ED simulation model in Figure 7 is data driven, scalable, flexible, and includes details such as state/time-dependent patient triage, legal nurse/patient ratio enforcement (according to California law) and all major equipment and staff. The Sigma engine automatically created from this model is run from an easy-to-use dynamic Excel spreadsheet interface. However, the point here is that the model was created by an outside simulation team (Berkeley undergraduates), and the modeling blocks are *connected*. Activity objects using RTMS do not need to be created by outsiders, and do not need to be connected by a simulation modeling team. Activity objects can be added or removed without destroying the logic of the simulation. Furthermore, by taking advantage of the fact that in the real world, many activities occur simultaneously, RTMS executes considerably faster than other commercial simulation software.

5 LESSONS LEARNED

One of the lessons learned (finally) in the meta-experiment with the two different approaches to modeling methodology research used here is that academics with new ideas should consider implementing them in commercial software, identifying their strengths, and using the marketplace to guide the research and development. This is a lesson first given to (and lost on) me by Alan Pritsker several decades ago when he left his professorship at Purdue (we were squash partners at the time) to form Pritsker and Associates, Inc. (P&A). P&A developed and integrated QGERT and GASP with continuous-time Systems Dynamics to create SLAM that was then rebranded as SIMAN and combined with CINEMA into ARENA, which is now being re-incarnated with 3-D animations as Simio.

Over the years, this has proven to be much more efficient path to improved simulation methodology than the academic research algorithm: (1) submit proposals until funded, (2) train graduate students, (3) present papers to peers, (4) hope a vendor will implement your methods, and (5) repeat indefinitely. It is a recurring theme of WSC panels why academic research has had almost no impact on simulation practice. Compare the two WSC panels a decade apart in Glynn et al. (1995) and Andridottir et al. (2005) where the issues discussed are almost identical (the author was on both panels). This perhaps is due to the fact that academics and software vendors have incompatible values: the first needs important unsolved simulation problems for research, and the second needs to convince their customers that they have solved all of them. Practitioners are trapped somewhere in the middle with less-than-optimal tools. This does not seem a problem in other fields like optimization or statistics – it is reasonable to ask why.

The other lesson is that radical notions like (S^3) need fundamental academic research as well as actual implementations beyond the toy models typically presented in journal papers. The two research environments are worlds apart, but their synergies are necessary for success.

ACKNOWLEDGMENTS

The author would like to thank Dr. Lenrick Johnston and David Zhang of Bio-G, inc., Dr. Aaron Bair of the UC Davis Medical School, Steven DeMello of Berkeley's School of Information and CITRIS, and the National Science Foundation's Small Business Innovative Research (SBIR) program for their support. Berkeley undergraduates, Shunpo Chang, Hac Phan, Majid Kahn, and Max Santana developed the term product that won a CITRIS grant for proof-of–concept integration of AI and IT into a self-simulating system at the UC Davis Emergency Department, whose staff has been welcoming and helpful.

REFERENCES

Andradottir S., D. Goldsman, B. Schmeiser and L. Schruben, and E. Yucesan. 2005. "Analysis Methodology: are we done?" In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 790-796. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Glynn. P. W., J. O. Henriksen, C. D., Pegden, B. W. Schmeiser, and L. Schruben. 1995. "The Interface Between Simulation Output Analysis Research and Practice," In *Proceedings of the 1995 Winter Simulation Conference*, edited by C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman, 346. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Miller, Martin J., David M. Ferrin, and Marcia G. Messer. 2004. "Fixing The Emergency Department: A Transformational Journey With Edsim." In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 1988-1993. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Mueller, R., C. Alexopoulos, and L.F. McGinnis. 2007. "Automatic Generation of Simulation Models for Semiconductor Manufacturing", In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 648-657. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Seila, Andrew, V. Ceric, and P. Tadikamalla. 2003. *Applied Simulation Modeling*. Duxbury Press.

AUTHOR BIOGRAPHIES

LEE SCHRUBEN is Chancellor's Professor at Berkeley and the Honorary (non-salaried) Chief Science (non-managing) Officer of Bio-G. Prof. Schruben holds degrees from Cornell, the University of North

Carolina, and Yale. He held the Schultz Endowed Professorship of Engineering at Cornell and is a Fellow of Informs, having served in every office of its Simulation Society. Email: LeeS@Berkeley.edu.