Towards Smarter Simulations:

How AI is Driving More Effective e-Learning

By James Ong and Randy Jensen

The use of computer-based simulations in training has a long history, dating back to the first aircraft flight simulators. Traditionally, these simulators emulated devices and physical laws, enabling students to learn cause and effect relationships, with a more economical and safe process than possible with live training and using actual devices. Continually evolving artificial intelligence technologies make possible even smarter simulations, dramatically improving training effectiveness. This article describes different types of intelligent simulation and how they are evolving.

History of Computer-based Training Simulations

The first training simulators emulated systems such as aircraft flight dynamics and controls, electrical circuits, ship turbine systems, and military weapons and battlefields. These simulations, behaving according to generally accepted causal relationships, enabled students to experience the effects of their actions and learned skills more safely, economically, and in many cases, more effectively than was possible via real-life training.

For example, Bolt Beranek & Newman developed the STEAMER[1] system in the early 80s to teach ship propulsion system principles to naval technicians. This innovative government-funded research system presented an interactive graphical display that enabled students to inspect and control the simulated propulsion system. Its carefully designed user interface presented a conceptually accurate (although not visually accurate) view of the system's components and their relationships, to help students acquire useful mental models of the system's operations. Although STEAMER was not artificially intelligent in the usual sense, its creators used what was then state-of-the-art AI development methods and technologies such as Lisp-based object-oriented programming languages and tools, Lisp workstations, and cognitive science theories regarding mental models.

Around the same time, Bolt Beranek & Newman employed AI programming languages and workstations to develop early versions of SIMNET, the first distributed, multi-player training simulation. Developed with Defense Advanced Research Projects Agency (DARPA) funding, SIMNET enabled networked simulators of the M1 Abrams main battle tank and the M2 Bradley infantry fighting vehicle to present crew training, collective command and control training, and maneuver training to Army personnel.

Intelligent Simulated Entities

Distributed military simulations allow students to coordinate their efforts with teammates while fighting opposing forces, sometimes in the presence of neutral actors. Unfortunately, requiring manual operation of all simulated positions was often undesirable, for several reasons. First, not all simulated positions offered equally useful training[2]. SIMNET and its successor training systems, primarily designed to teach U.S. military tactics, presented a less valuable learning experience to those assigned to controlling enemy forces manually. Second, when a simulation exercise involved many different types of actors, each requiring specialized skills, it was logistically difficult to assemble the necessary complement of people and skills. Ideally, a student could practice his or her skills in a simulation comprised of human-controlled and computer-generated actors, depending upon the number and types of other students desiring training at that time.

As soon as distributed simulations appeared, there was an appreciation for the value of intelligent simulated entities. Enhancements to SIMNET provided Computer-Generated Forces (CGF) that emulated enemy tactics and doctrine reduce the number of people required to control opposing forces, lowering the cost and difficulty of conducting large-scale simulated training exercises.

Although the technology of computer-generated forces advanced steadily, thanks to a succession of government-funded research and development programs, challenges remain. Achieving more realistic and capable simulated behaviors is fundamental, and some artificial intelligence researchers are applying AI methods and tools to address this problem. For example, Soar is a model of human cognition that is recognized by programming tools and methods that support operational AI systems development. Soar implements its permanent knowledge as production rules and incorporates additional features to improve its power and efficiency. For more information on Soar, visit: *http://ai.eecs.umich.edu/cogarch0/soar* or *http://www-2.cs.cmu.edu/afs/cs/project/soar/public/www/home-page.html.*

Simplifying the process of specifying intelligent behaviors is another challenge. Subject matter experts, not programmers, understand the best algorithms to control the simulated entity behavior. However, traditional language-based methods for specifying behaviors require programming skills experts do not often have. Therefore, experts must describe the desired behaviors to software engineers who then translate these behaviors into software. This multi-step process is expensive, time-consuming, and error-prone. To simplify the creation of sophisticated and realistic behaviors within simulations and games, AI software engineers at Stottler Henke developed an authoring tool and run-time engine called SimBionic[3] that enables experts to define executable behaviors graphically, without programming. These behaviors, encoded as hierarchical finite state machines, communicate with one another via blackboards.

Intelligent Instructional Agents

Training simulations help students learn by presenting *realistic, natural feedback* in response to their actions. However, simulations of complex systems or situations often make it difficult for students to determine what they did correctly or incorrectly based on the final results. For example, even if a student or team of students won a game or battle it does not mean that they did everything correctly. A student may achieve the desired goal within the simulation without realizing the approach was inefficient, unreliable or even incorrect. Therefore, complex simulation exercises are often observed by instructors who provide coaching and hints, assess student performance, facilitate discussion and reflection and offer critiques and feedback.

Ideally, each student would receive the personal attention of a tutor or coach who evaluates each action carried out by the student, identifies the student's strengths and weaknesses, and offers appropriate hints, feedback, and learning challenges. However, supplying one-on-one instruction for every student is too expensive. Artificial intelligence technologies augment simulations to create *Intelligent Tutoring Systems*[4] that offer individualized instruction automatically and cost-effectively. These systems, often called *cognitive tutors*, employ cognitive models of student learning.

These systems incorporate three types of knowledge:

What to teach – knowledge of the subject area or task that enables the tutor to assess the student's knowledge and skills, based on the student's actions within the simulator.

Who is being taught – estimates of the student's knowledge, skills, and other attributes, based upon the student's performance using the simulator and other information.

How to teach – instructional strategies that control how the tutor selects or generates hints, feedback, questions, or learning challenges that are appropriate for the student.

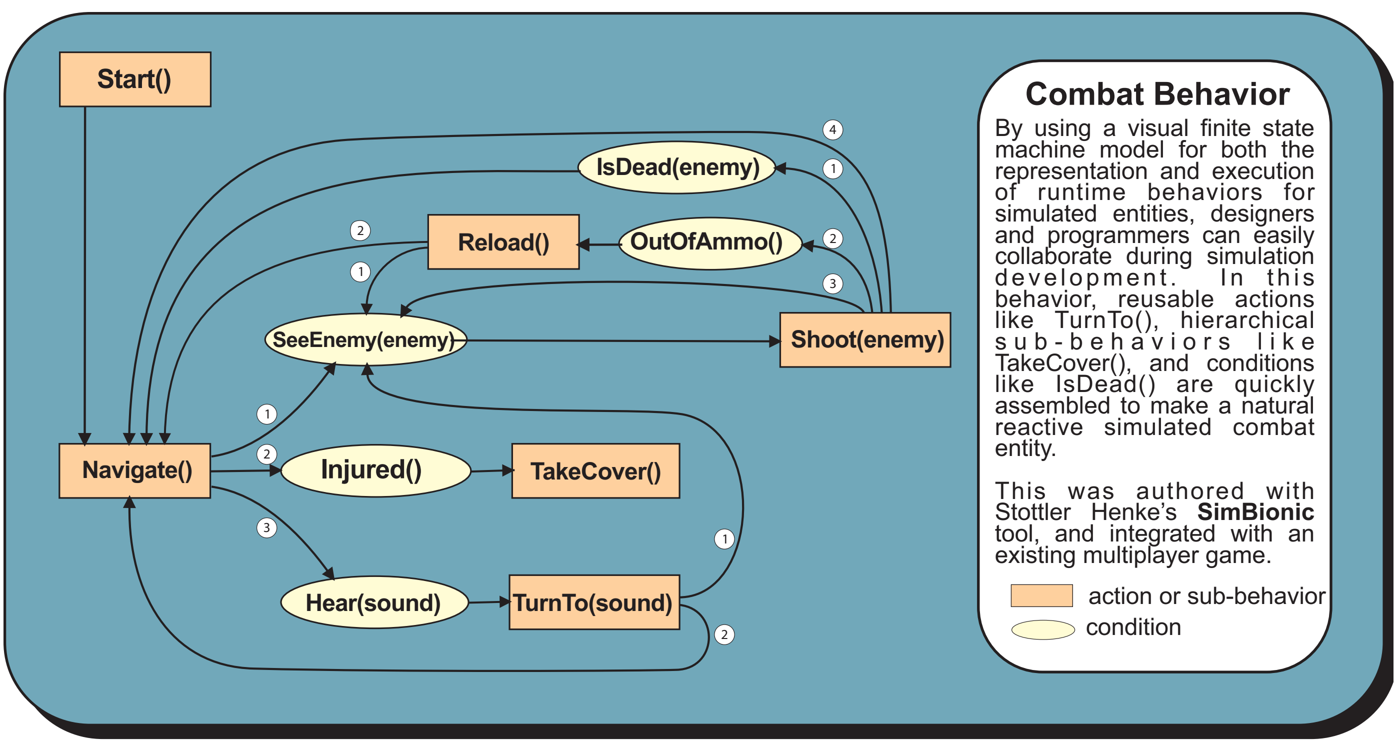

Figure 1: Combat Behavior

Tutoring systems that evaluate student actions and assess strengths and weaknesses use different AI approaches. Model-tracing tutors such as the algebra, geometry, and Lisp tutors developed at Carnegie Mellon University employ an expert system (often, rule-based) capable of solving any problem generated by the tutor and presented to the student. The tutor evaluates the student's actions by comparing them with the solution(s) generated by the expert system. This approach has the advantage of using an expert system to solve any problem within its domain and enables the tutor to generate and present a large number of possible problems to the student. However, this approach requires the potentially difficult and expensive development of an expert system.

By contrast, case-based tutors augment general domain knowledge with scenario-specific knowledge. Using this knowledge, the tutor infers intentions from the student's actions and determines their appropriateness, even in complex domains, by taking advantage of the context imparted by each scenario. However, this approach often requires the development of a domain-specific authoring tool or knowledge acquisition tool to enable course developers or subject matter experts to easily create scenarios and enter scenario-specific assessment and tutoring strategies.

In many subject areas, there is no single correct solution to most problems — students must apply their knowledge and skills to weigh competing goals and considerations. Socratic tutoring systems help students master these subject areas. They do this by conversing with the students to point out facts, ask questions, or present hints and feedback, and prompting them to explain and reflect upon their actions, beliefs, and reasoning. Development of automated Socratic dialogs relies on two AI technologies:

Natural Language Understanding enables the tutor to interpret the student's free-text utterances and replies. Although it is generally very difficult to create parsers that understand unrestricted natural language, Socratic tutors can exploit expectations created by the dialogs current context to restrict or bias the set of possible interpretations of the student's utterances, increasing the parser's accuracy.

Automated planning enables the Socratic tutors to generate appropriate statements and questions to pursue instructional goals. These are a result of the tutor's assessment of the student's learning objectives and needs. These instructional planners can use a combination of generic, domain-specific, and scenario-specific tutoring strategies.

Emotionally Realistic Agents

Many important skills involve interacting with other people.. To offer a useful learning experience for these skills, social simulations let students interact with realistic, simulated characters. These characters are controlled by computer models that specify how their actions and internal mental states (knowledge, beliefs, emotions) are affected by the student's actions and other simulated events.

For example, the Applied Physics Laboratory at Johns Hopkins University developed a simulation that enables law

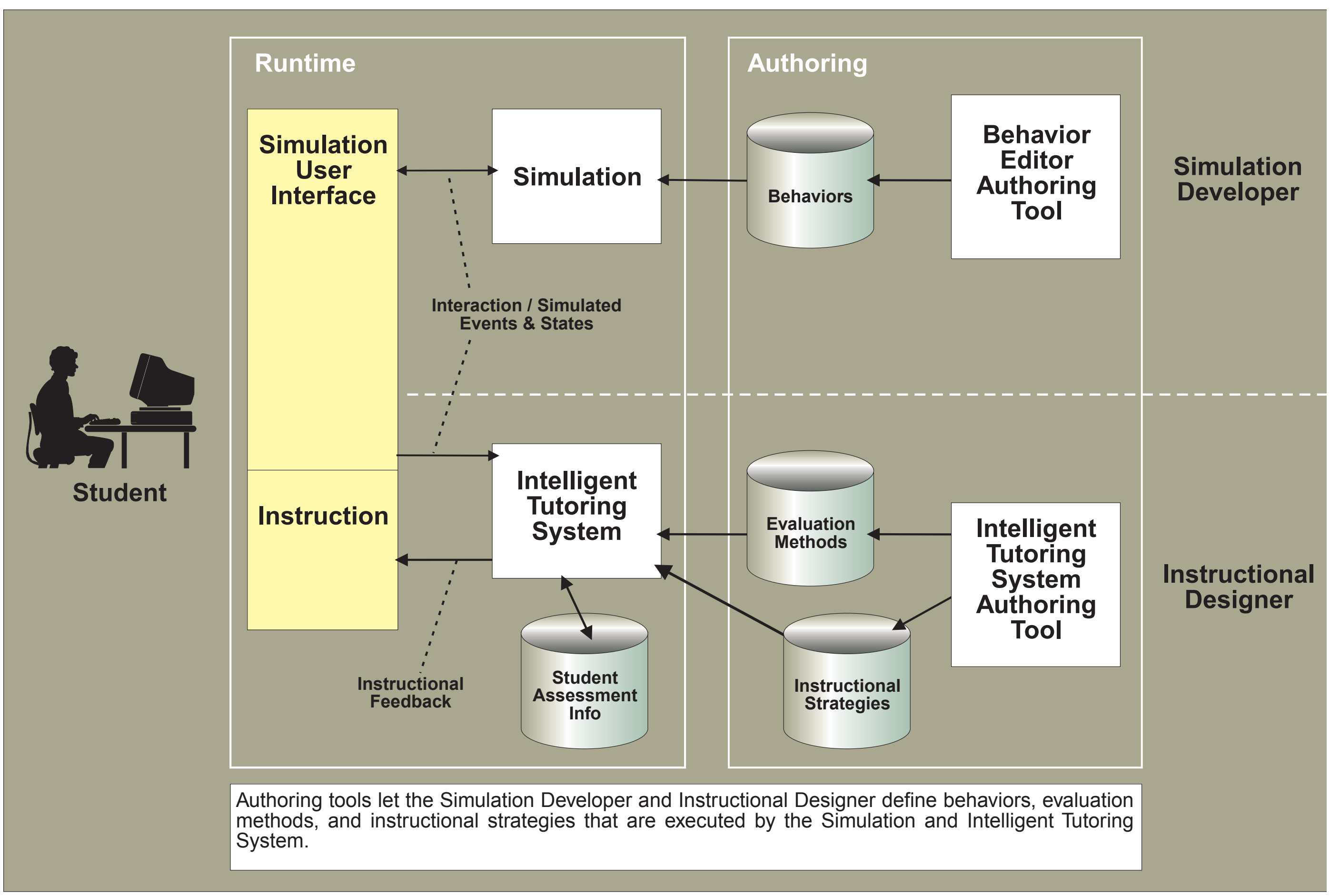

Figure 2: Authoring and run-time environment.

enforcement personnel to practice interrogating simulated subjects[5]. Although each student's primary goal is to obtain information from the subject, successful interrogation requires the student to pursue competing sub-goals such as establishing and maintaining rapport, controlling the discussion topic, and phrasing questions appropriately. During each scenario, the simulated subject responds to the student's statements and questions in an emotionally-realistic manner. This response is based on what the interrogator said and on the model of the subject's mental and emotional state. Another social simulation example is the Yello program, developed at Northwestern University[6]. This system enables students to sell Yellow Pages advertising space during simulated sales calls with prospective customers.

Emotionally Aware Agents

Knowing the student's mental and emotional state helps the intelligent agent assess or instruct the student more effectively. For example, a student might perform poorly because the task is simply too difficult. On the other hand, the poor performance may be due to stress, fatigue, cognitive overload, a misconception, low motivation, or lack of attentiveness. Before the instructor can address the student's poor performance effectively, he or she must identify the underlying cause.

Unfortunately, most techniques used by human instructors to assess their student's mental and emotional state are difficult or expensive to implement within a computer-based system. For example, interpreting facial expressions or body language requires sophisticated computer vision capabilities to see and interpret subtle visual cues. Making inferences from the wording and timing of the student's utterances requires very advanced speech recognition and analysis of natural language. On the other hand, sensors that monitor the student's physiological state, such as heart rate, skin conductance, pupil dilation, and voice quality can, in theory, be combined using data fusion techniques to offer useful clues.

Tutors can also use an improved understanding of the student's personality, learning style, and other attributes to select the best instructional approach. For example, some students stay motivated when they experience frequent positive reinforcement, so the tutor supplies these students with repeated opportunities to apply skills they have already mastered. However, other students fare better when presented with difficult challenges. While some students prefer an introduction to a new subject area via a systematic presentation of facts and general concepts, other do better when presented a specific scenario and goal that defines a context and motivation for learning the material. The challenge is identifying reliable methods for assessing these individual differences and using these assessments to select the best instructional method. This is a current research area in cognitive psychology and AI systems development.

Conclusions

Over the past several decades, artificial intelligence and training simulations have enjoyed a symbiotic relationship. AI supplied techniques and conceptual frameworks that improved the effectiveness of training simulations by making them more intelligent, realistic, and engaging. Training simulations offered challenging application areas with a range of AI technologies applied and extended to create effective intelligent simulated entities and intelligent tutors. With much of the early funding coming from the government, much of the early research and development targeted improved military training. However, many practical intelligent simulation technologies have become available for teaching soft skills (sales, communications, leadership, teamwork) and technical skills (using software, operating equipment, or executing organizational processes) within corporations.

References

1. Steamer, James D. Hollan, Edwin L. Hutchins, Louis Weitzman STEAMER: An Interactive Inspectable Simulation-Based Training SystemAI Magazine5(2): Summer 1984, 15-27.

2. Military simulation and training *www.mt2-kmi.com*

3. SimBionic *www.simbionic.com*

4. Intelligent tutoring systems, *www.aaai.org/AITopics/html/tutor.html*

5. Johns Hopkins FBI interviewing simulation, *www.jhuapl.edu/education/sciencetech/factpctrain.html*

6. Yello sales simulation, *www.engines4ed.org/hyperbook/nodes/NODE-302-pg.html*

James Ong is a Group Manager at Stottler Henke Associates, Inc., an AI software development and consulting firm. Mr. Ong leads teams developing intelligent systems for education and training, automated scheduling, automated system diagnosis, and computer-mediated team collaboration and coordination. He holds degrees from Boston University, Yale University, University of California at Berkeley, and the Massachusetts Institute of Technology. He can be reached at *ong@stottlerhenke.com* or 650.655.7242.

Randy Jensen is a Project Manager for Stottler Henke Associates, Inc., and has developed intelligent simulated entity behaviors and tutoring systems for a variety of domains. He has a BS in Symbolic Systems from Stanford University.

Other Sources of Intelligent Tutoring Related Information

STATUTOR: Intelligent Tutoring System? A paper describes issues that arose from the development of an Intelligent Tutoring System to educate students in the statute law domain. (1996). (*http://elj.warwick.ac.uk/jilt/BILETA/1996/3hegarty/default.htm*)

Web-Based Intelligent Tutoring Systems. Sherry L., PC AI 14.2 Mar/Apr 2000, He examines how AI techniques are incorporated in web-based intelligent tutoring systems.

Strategies for Intelligent System Development, Sherry L., PC AI, 12.6 Nov/Dec 1998, Describing a statistical model that measures the performance of a development team.

CIRCSIM-Tutor Project, Illinois Institute of Technology Dept. of Computer Science and Rush College of Medicine Dept. of Physiology, The CIRCSIM-Tutor project is building a language-based intelligent tutoring system for first-year medical students to learn about the reflex control of blood pressure. Students solve small problems and are tutored by Socratic dialog with the computer. *www.csam.iit.edu/~circsim*

Intelligent Tutoring System A learning technology system that dynamically adapts learning content to a learner's specific objectives. *http://ltsc.ieee.org/doc/wg3/interactive_glossary/ITS.htm*

Student Modeling in an Intelligent Tutoring System for the Passive Voice of English Language (Educational Technology & Society) *http://ifets.ieee.org/periodical/vol_4_2000/virvou.html*

Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando. Delivering Hints in a Dialogue-Based Intelligent Tutoring System, *www.csam.iit.edu/~circsim/documents/yzaaai99.pdf*

Adding an Intelligent Tutoring System to an Existing Training Simulation Richard H. Stottler and Randy Jensen Stottler Henke Associates, Inc. *www.shai.com/papers/IITSEC-02-BC2010.pdf*

Intelligent Tutoring Systems - Explores intelligent tutoring systems with an emphasis on experimental evaluation and field trials. *www.acrc.unisa.edu.au/is/its*

Knowledge Based Systems in Adaptive Training: Tailoring Training to the Individual, Brosch B, PC AI 16.4 2002. He demonstrates the use of a knowledge based system to construct assessment driven systems with self-customizing tutorials.

First Neural Engineering Conference Highlights Neuroscience Advances

First International IEEE Engineering in Medicine and Biology Society (EMBS) Conference on Neural Engineering. It will be held 20–22 March, 2003, at the Grand Hotel Quisisana on Capri Island, Italy.

This conference will include presentations about advancements in engineering technologies, including micro- and nanotechnology image and signal processing and algorithms.

The scientific program will cover nine areas: the brain and neurons; artificial implants and neural prostheses; biological neural networks; control of neurological systems; neural signal processing; neural informatics; brain imaging; the brain-computer interface; and virtual and augmented reality in brain surgery, diagnosis, and treatment.

www.dartmouth.edu/~ne2003