

Capturing and Building Expertise in Virtual Worlds

Jared Freeman, Webb Stacy,
Jean MacMillan, and Georgiy Levchuk

Aptima, Inc., 1726 M Street, NW, Suite 900, Washington, DC 20036, USA
{jfreeman, wstacy, jmacmillan, glevchuk}@aptima.com

Abstract. Model-driven simulation can make the design and delivery of instruction more efficient and effective. We describe two computational models that support both the design and delivery of instruction. BEST (the Benchmarked Experiential System for Training) can guide experts through the space of domain problems during the knowledge engineering phase of instructional design; it can guide trainees through the space of training objectives during instruction. PRESTO (Pedagogically Relevant Engineering of Scenarios for Training Objectives) builds scenarios on the fly to elicit the knowledge of experts during instructional design, and to satisfy the instructional objectives of trainees.

Keywords: Adaptive instruction, knowledge engineering, Constraint Logic Programming, Markov Decision Process.

1 Introduction

Training organizations often marry 21st century simulation technology to 20th century methods of designing instructional content and training scenarios.

Instructional design, which defines the objectives and content of training, engage scarce experts in interviews and qualitative analyses that are so time-consuming they are derided as a bottleneck [1].

Scenario design in simulation-based training creates artworks – engaging but singular products that present fundamentally the same instructional experience to all students. Rarely are training scenarios developed in libraries that systematically vary factors related to training objectives, and rarely can simulators adapt scenarios on the fly to respond to the instructional needs of each student. This is true, in general, whether we conceive of scenarios as brief vignettes (e.g., a specific air-to-air engagement) or as the larger events in which they are combined (e.g., a mission of many engagements), and whether the training simulator serves a one trainee (e.g., a game) or many (e.g., a military exercise with live, virtual, and constructive components).

We are exploring ways in which simulations can be mated to computational models that specify and construct scenarios to meet their users' needs. This strategy has value during scenario design; it configures training scenarios to the needs of multiple students. The technologies also have value during instructional design, where they could improve the capture and analysis of expertise.

We open this paper with a brief discussion of expertise. It is what instructional designers seek and what scenario designers hope to deliver to students through their simulations. In particular, we develop a spatial metaphor for expertise, and we use this to describe how two modeling technologies that we have implemented automate scenario design and could, potentially, automate instructional design.

2 Expertise

Expertise is the ability to discriminate meaningful classes of domain features and patterns, and to take decisions or actions that are appropriate to the class at hand.

This conceptualization of expertise is illustrated in Figure 1, which depicts the novice’s conception of a domain as an undifferentiated space of objects and events. In contrast, the expert’s understanding is structured to distinguish functionally distinct and important objects, events, and situations. Expert partitioning of the domain space supports decision making and action. Each partition of the space is associated with distinct issues for consideration in decision making, or with unique actions such as tactics.

Studies of physicists, software engineers, chess masters, and others validate that domain knowledge is more functional (useful) among experts than novices, and more uniform. For example, expert physicists sort physics problems by the deep structure or functions they represent (e.g., acceleration), while novices sort by surface features such as the materials (e.g., balls rolling down hills) cited in the problem [2]. Expert software engineers sort programming terms into groups that are similar to those of other software experts and are relevant to programming, while less expert engineers are more varied in their classification schemes and those schemes are often based on everyday meanings of the terms [3]. Chess masters reconstruct briefly viewed chess boards in tactically important clusters of pieces (e.g., defensive and offensive clusters), while novices rebuild boards in an order that is arbitrary, with respect the strategies of chess [4]. In short, expert knowledge of domains is partitioned in ways that reflect meaningful differences between problems, differences that bear on the accuracy of decisions and actions.

The concept that expert knowledge is functionally structured has two ramifications for the design of scenarios or their component vignettes. First, we should create the range of scenarios that represent each partition of the domain, so that students can experience the challenges they pose. Such scenarios must represent the training objectives students need to address, the conditions (e.g., events) under which students must learn to perform, and the measures with which

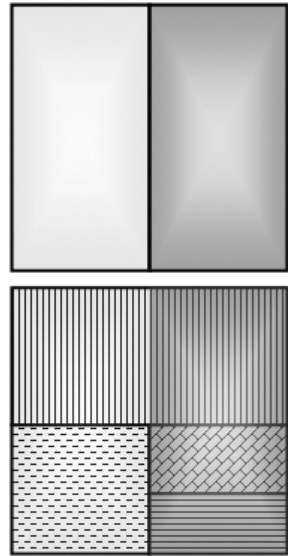


Fig. 1. Novices perceive and respond to surface features of domain problems (shading, above). Experts detect underlying problem structure (patterns, below), which maps well to correct solutions.

trainers evaluate student expertise. Second, we must guide students from scenario to scenario (partition to partition) so that they learn to distinguish between the problems each presents, and respond to them as experts do.

This concept of expertise also has implications for using simulations as knowledge engineering tools during instructional design. First, we can build scenarios (or vignettes) that represent the full range of domain problems, to enable experts to demonstrate their capabilities. Second, we can guide experts through that space in a way that enables us to efficiently discover the bounds experts draw, the partitions. This capability is most useful in domains in which experts are known to exist, in which the structure of expert domain knowledge is unknown or contested, and actions or tactics are not well mapped to that structure.

We are developing two modeling technologies that address these opportunities to improve training. To deliver scenarios that meet the specific needs of each student, BEST (the Benchmarked Experiential System for Training) guides students through the space of instructional objectives and corresponding scenarios; this demonstrably accelerates the development of expertise. PRESTO (Pedagogically Relevant Engineering of Scenarios for Training Objectives) adapts scenarios on the fly to satisfy instructional objectives or specifications, such as those BEST recommends. These models also have the potential to break the bottleneck of knowledge engineering. BEST can pilot experts efficiently through a potentially vast space of domain problems, which PRESTO then composes in scenarios on demand. The clusters of problems to which experts respond similarly each define a partition, a subspace of functionally similar domain problems. Instructional designers, having discovered these partitions, can rapidly develop didactic training that emphasizes the characteristics that distinguish each partition (or cluster of problems). They can transform the scenarios used in knowledge engineering into training scenarios that give novices a truly representative sample of experiences in the domain. We describe each of these technologies, below.

3 Scenario Specification and Sequencing with BEST

The Benchmarked Experiential System for Training (BEST) specifies the training treatment that a simulation should present next to advance a student farthest towards expertise.

To accomplish this, BEST uses a Partially Observable Markov Decision Process (POMDP) model to represent our probabilistic knowledge of the level of student competency (in multiple dimensions), and the probabilistic effects of training treatments on these competencies. These training treatments may be expressed as training objectives (“The student should acquire skill X, next”), training conditions (“Increase the challenge from enemies.”), or training scenarios (“Present scenario 127 next”). Their instructional effects of each treatment on students of different abilities can be estimated by experts or learned from training performance data. BEST computes an optimal training policy [5] that adapts over time to specify next most beneficial simulation scenario given the trainee’s most recent as well as history of performance.

In two experiments to validate BEST as a training adaptation tool, teams of subjects received simulator training in the complex tasks of the Air Operations Center

Dynamic Targeting Cell. The training experiences were drawn from a large library of approximately 50 scenarios that varied systematically on the intensity of defensive and offensive challenges. The instructional strategy for selecting from this library was controlled either by the BEST POMDP or by a hierarchical part-task training scheme that advanced trainees through increasing numbers of targets (offensive challenges), and then increasing numbers of threats (defensive challenges). The BEST solution reliably increased learning relative to the control condition, holding the number of training trials constant (see Figure 2) [6].

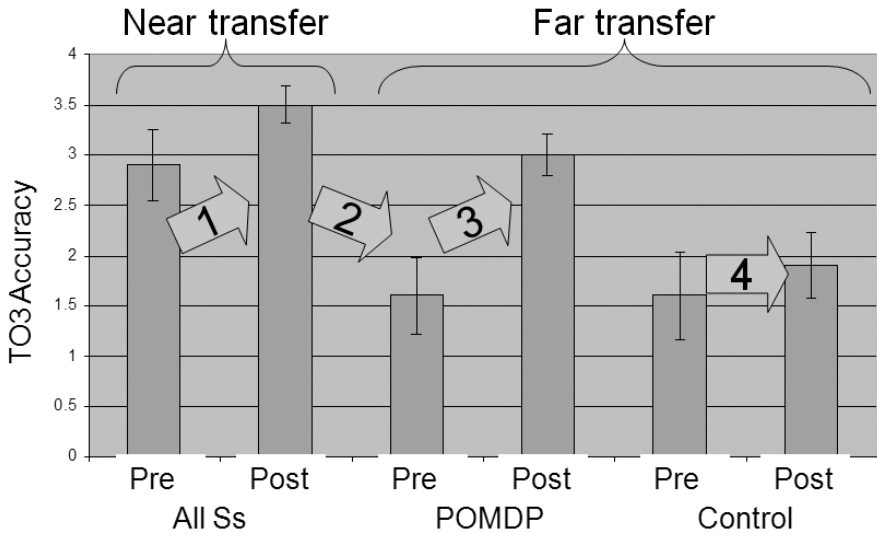


Fig. 2. Experimental tests of BEST found that teams learned the complex task of the Air Operations Center Dynamic Targeting Cell ($p < .01$); that far transfer within the task degrades performance ($p < .01$); that, on far transfer problems, teams in hierarchical part-task condition (control) do not reliably learn ($p > .05$), while those in the BEST / POMDP condition do ($p < .01$)

In the same way that BEST navigates a space of existing scenarios or training objectives, it can be used to learn the structure of an ill-defined domain space by observing expert behavior. This application of BEST, which we plan to attempt in the near future, requires that a large library of scenarios be constructed or generated on the fly (see PRESTO, below). These scenarios would sample the space of domain conditions randomly (at worst) or systematically according to some theory about important classes of domain problems or features. BEST would apply policy learning techniques to (1) formal descriptions of the features of each scenario and (2) expert decisions or actions in the scenarios. It would discover those partitions of the space (clusters of scenarios) to which experts generally respond in a uniform manner. Each such partition in the domain space consists, by definition, of functionally equivalent problems. In the terms of a Markov Decision Process, BEST would discover the groups of scenarios that constitute each unique “state”, and it would discern the “action” that experts apply to the state to address the challenge it poses.

This application of BEST could be used to confirm theories about how experts understand and act in complex domains. In air-to-air warfare, the tactics of enemies are well known and our responses are well documented. However, experts make subtle distinctions in their interpretation of enemy approaches and their own application of tactics; model-driven simulations might efficiently discover these distinctions. BEST may have greater value in domains of human performance that are not well understood. It might, for example, be used to analyze how experts inspect multi-spectral imagery, how they monitor crowds to identify potential insurgents, and how they scan roadways for Improvised Explosive Devices. In all of these cases, experts perceive configurations of the environment as innocent, suspicious, or threatening; the take actions to test their perceptions; and they may take actions to address threats. To train the next generation of experts, we must analyze how today's experts understand (i.e., partition) the problem space in the domain, and how they choose their actions. Model-driven simulation should make this analysis much more thorough and efficient than traditional interview methods alone.

4 Scenario Construction with PRESTO

PRESTO (Pedagogically Relevant Engineering of Scenarios for Training Objectives) builds or revises simulation scenarios to ensure that they present the conditions required to meet specific training objectives [7, 8].

PRESTO accomplishes this by formally representing training objectives, domain objects, and domain events. PRESTO applies constraint logic programming (CLP) techniques to (re)formulate the schedule of training objectives and events addressed by a scenario. PRESTO defines temporal relations between domain events, spatial relations between domain objects that are required for training events, and other necessary scenario preconditions, so that it is possible for students to address training objectives (see Figure 3). It is particularly useful in simulations in which there are many students with potentially conflicting training objectives, all operating in a single, complex environment.

For example, suppose that a helicopter crew is training on a sensor fusion task. Performance measures indicate that they need additional work when certain combinations of targets are present. However, this combination occurs only once in the original scenario, and that occurrence was fouled when the crew unexpectedly chose a course that took them far away from the sensor targets, early the training scenario. In mid-exercise, PRESTO can determine whether it is feasible to add more instances of this target combination, and determine when and how to do so, given the constraints imposed by satisfying all other training objectives for all trainees engaged in the scenario. In reworking the scenario, PRESTO takes care not to interfere with the ongoing training of other students.

PRESTO is designed to support training delivery, per the description above. However, we hope to apply it to facilitate knowledge engineering as well. PRESTO can represent theories about the classes of problems experts perceive in a domain, and generate scenarios that present each class. When expert actions are relatively uniform across a group of scenario events (i.e., problems), a coherent partition of the space has

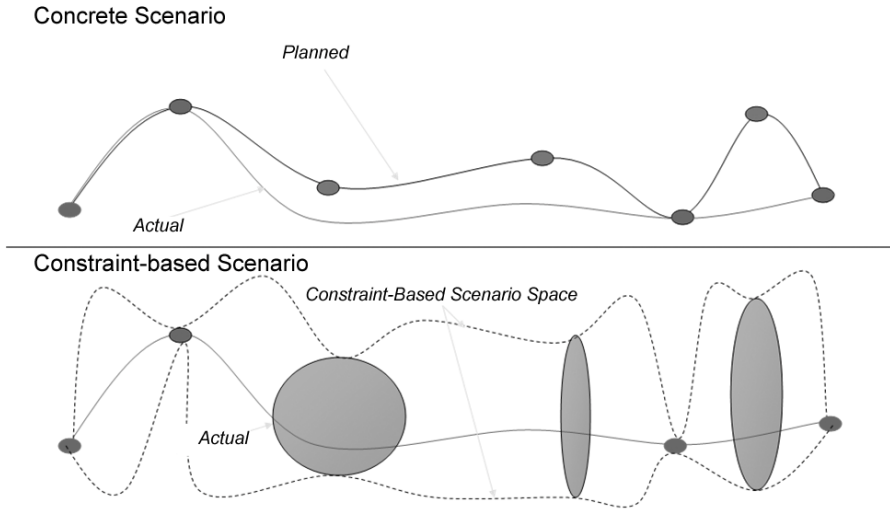


Fig. 3. Simulation scenarios are traditionally designed to draw trainees through a specific series of instructionally critical events (above), but these plans are often foiled by the actions of other trainees or system failures. PRESTO effectively models the bounds (constraints) of the simulation space and generates scenario events within those bounds to meet training objectives (below).

been discovered. When expert actions vary between scenarios, they represent meaningfully different problems or partitions of domain knowledge. PRESTO, in short, can generate representative samples of domain problems in a controlled manner, so that analysts can learn which problems are substantively different to experts.

5 Conclusion

We have described two computational models that can drive simulations to improve training delivery and knowledge elicitation. BEST specifies the training objective or class of problem from which a student can glean the greatest instructional benefit. PRESTO constructs scenarios that present such problems. These models were developed to ensure that simulators adapt to trainees, whether by selecting the best scenario among a library of many, or generating a scenario customized to the trainee. These technologies have potential, also, to accelerate knowledge engineering during instructional design. They should, if adapted to that task, help define and generate the content for training in new and complex domains. Model-driven simulation has the potential to make instructional design and scenario design more systematic and effective. This should greatly increase the impact of simulation-based training.

Acknowledgments. The research and development reported in this article were funded in part by the Department of Defense. We thank Melissa Walwanis of NAVAIR / Orlando for supporting development of PRESTO for automated scenario generation. We thank the Jerome Busemeyer and Robert Sorkin, formerly of the Air

Force Office of Scientific Research, for supporting the development of BEST algorithms and the empirical validation of them, which we conducted with Wayne Shebilske of Wright State University.

References

1. Rose, C.P., Gaydos, A., Hall, B.S., Roque, A., VanLehn, K.: Overcoming the Knowledge Engineering Bottleneck for Understanding Student Language Input. In: Proceedings of AI in Education. IOS Press, Amsterdam (2003)
2. Chi, M.T., Feltovich, P.J., Glaser, R.: Categorization and representation of physics problems by experts and novices. *Cognitive Science* 5, 121–152 (1981)
3. Cooke, N.J., Schvaneveldt, R.W.: Effects of computer programming experience on network representations of abstract programming concepts. *International Journal of Man-Machine Studies* 29, 407–427 (1988)
4. Chase, W.G., Simon, H.A.: Perception in Chess. *Cognitive Psychology* 4(1), 55–81 (1973)
5. Levchuk, G.M., Gildea, K., Freeman, J., Shebilske, W., Alakke, G., Narakesari, S.: A Benchmarked Experiential System for Training (BEST) for Optimizing Instruction. In: Proceedings of the 12th International Command and Control Research and Technology Symposium, New Port, RI (2007)
6. Levchuk, G., Shebilske, W., Freeman, J.: A model-driven instructional strategy: The benchmarked experiential system for training (BEST) (in review). Submitted to *Journal of Mathematical Psychology*
7. Stacy, E.W., Walwanis Nelson, M., Colonna-Romano, J.: Using Pedagogical Information to Provide More Effective Scenarios. In: Proceedings of the 2007 Interservice/Industry Training, Simulation, and Education Conference, Orlando, FL (2007)
8. Stacy, E.W., Colonna-Romano: From events to experiences: Focusing scenarios on training effectiveness. In: Walwanis Nelson, M., Lowe, M. (eds.) *Advances in the Theory and Practice of Scenario-Based Training—Charting the Road Ahead*. Proceedings of the 2008 Conference on Behavior Representation in Modeling and Simulation, Providence, RI (2008)