Toward Automated Scenario Generation with GIFT

J. T. Folsom-Kovarik[1], Jonathan Rowe[2], Keith Brawner[3], and James Lester[2]
Soar Technology, Inc.[1], North Carolina State University[2],
U.S. Army Combat Capabilities Development Command – Soldier Center – Simulation and Training Technology Center[3]

Introduction

Adaptive instructional systems such as the Generalized Intelligent Framework for Training (GIFT) can tailor training to meet the learning needs of individuals and teams. A significant cost driver in the design and development of adaptive instructional systems is the manual creation of training scenarios. Delivering personalized instruction to students requires the creation of a broad range of instructional materials. Without effective automation, the tailoring that adaptive instructional systems implement is limited by the small number of instructional variants that a human author can define, as well as a one-size-fits-all approach to training. Further, additional scenarios are useful for enhancing replay through drill-and-practice of specific skills. Generating training scenarios for adaptive instructional systems includes two key components: (1) creating novel scenario content, and (2) devising models that dynamically tailor scenario content to learners.

This chapter discusses two parallel efforts to enhance GIFT through the design, development, and investigation of *automated scenario generation*. First, we describe a scenario variation tool that creates many variants of training scenarios to offer the instructor (or GIFT) increased choices between different combinations of instructional support or challenge. Second, we describe a data-driven framework for dynamic scenario adaptation that models how simulation-based training scenarios can be tailored at run-time to foster optimal learning outcomes. These are two possible approaches to addressing the authoring bottleneck inherent in adaptive instructional systems.

In the first approach, the scenario variation tool makes use of a novelty search algorithm (Lehman & Stanley, 2008, 2011). Genetic algorithms such as novelty search rely on a population of prospective solutions which are modified with 'mutation' or 'crossover' operations to create new prospective solutions in a repeating cycle. Prospective solutions with maximum fitness survive and reproduce in the population from one cycle to another. Novelty search replaces the typical genetic algorithm fitness evaluation with a novelty evaluation (Gomes, Urbano, & Christensen, 2012). In this replacement, genetic variants do not compete to become better, but to become *different*. Novelty search has already been used with success to evolve content similar to training scenarios, such as game levels (Liapis, Yannakakis, & Togelius, 2015). In the present research, training scenario variants attempt to become different as measured by the support or challenge they offer learners. As a result, novelty search is well suited to expand the space of possible training scenarios that GIFT can choose from when it tailors training (Folsom-Kovarik & Brawner, 2018). The scenario variations that result from novelty search provide varying levels of support or difficulty, such as offering a series of increasingly more complex scenarios, varying scenario events while ensuring that complexity is comparable, and offering scenarios that combine more complexity in one learning objective but less complexity in another learning objective that requires support.

The second approach being investigated, the dynamic scenario adaptation framework, DeepGen, utilizes reinforcement learning (RL) to induce models for run-time tailoring of training scenarios to achieve instructor-specified learning objectives (Rowe, Smith, Pokorny, Mott, & Lester, 2018). RL refers to a family of machine learning techniques for solving tasks involving sequential decision-making under uncertainty (Sutton & Barto, 2018). Over the past several years, a range of RL techniques have been investigated for run-time personalization of virtual learning environments for K-12 and undergraduate education, including modular RL (Rowe & Lester, 2015), multi-objective RL (Sawyer, Rowe, & Lester, 2017), constraint-based RL (Shen, Mostafavi, Barnes, & Chi, 2018), inverse RL (Rafferty, Jansen, & Griffiths, 2016), and deep RL (Wang, Rowe, Min, Mott, & Lester, 2018). Building upon this foundation, DEEPGEN utilizes RL to induce models for enacting run-time adaptations to military training scenarios, aiming to produce training experiences that optimize learning outcomes or provide effective assessments of target skills. Rather than novelty being the primary selection mechanism of scenario selection, as above, the RL-based dynamic scenario adaption uses a population of simulated students.

This chapter is organized as follows. In the next section, we describe work investigating novelty search and RL, respectively, to automatically generate training scenarios in two military training domains. We then describe efforts to engage subject-matter experts to obtain feedback on how to deliver automated scenario generation capabilities to instructors and scenario developers. Next, we discuss initial findings from the two projects, and conclude by offering recommendations for GIFT and directions of future work.

Scenario Generation Methods

Two complementary demonstrations of the two approaches in two domains of militarily-relevant instruction were carried out to investigate automated generation of training scenarios. First, novelty search was demonstrated in a small unmanned air system (SUAS) training scenario. Second, reinforcement learning-based scenario adaptation was demonstrated in the domain of artillery call for fire (CFF) training.

Novelty Search to Automate SUAS Training

The first demonstration focused on generating many variants of training scenarios in advance of training. Infantry employment of small unmanned air systems can be trained with a scenario structured by information delivery and choice points. Trainees work through the process to plan, prepare, and execute a UAS mission by making decisions based on information such as mission briefings, UAS observations, and popup events. Optimal and acceptable decisions continue the scenario to the next choice point, while one or more unacceptable decisions can cause remediation and restart. In this setting, novelty search can offer GIFT valuable opportunities to change scenarios after a restart or to challenge different aspects during training based on learner choices or GIFT's internal learner model.

Technical demonstrations of novelty search showed the technique can generate hundreds or thousands of scenario variants, and the variants are measurably different or similar enabling fine-grained matching to instructional needs (Figure 1). GIFT could match generated scenarios to learners' needs via 48 measures, scaled continuously between support and challenge, representing the 48 learning objectives (LOs) covered in the original training system. The variations did not control LOs directly, but controlled the number, size, and location of units and areas anywhere on a scenario map. A single variation in one such element might alter the scenario's support for many LOs. For example, moving an enemy armor unit might challenge a recon LO if the unit moved

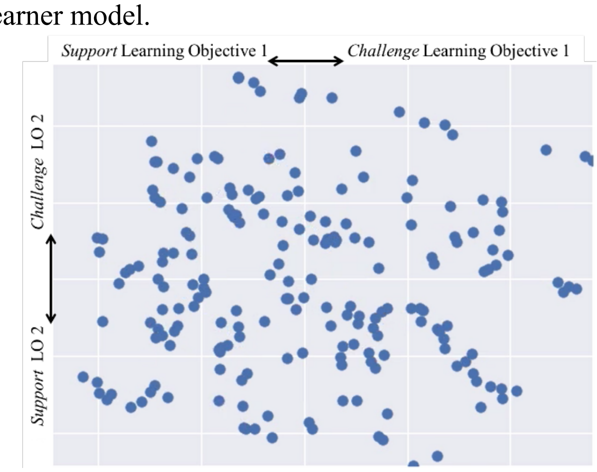

Figure 1: Novelty search creates training variants with many combinations of support and challenge.

into a wooded area and simultaneously support a dynamic response LO if the tank moved further away from friendly forces. As a result of these interactions, novelty search tended to find scenario variations that supported some LOs and challenged other LOs in combinations that had not previously existed.

The authors explored several methods to present the wealth of variations to nontechnical end users, such as instructors who wish to review the available variants or select one training variant which will best support specific trainees. An initial evaluation by a subject matter expert (SME) resulted in defining a presentation of training variants that military instructors are likely to find valuable. The initial evaluation resulted in user interface recommendations to translate technical variation into a human-usable presentation, and will support an upcoming evaluation by operational users.

Reinforcement Learning-Based Scenario Adaptation in Call for Fire Training

The second demonstration focused on devising RL-based policies for adapting events in example CFF training scenarios. In a CFF mission, an infantry soldier requests indirect fire on a target from supporting artillery (e.g., mortar, field artillery, unmanned aircraft). The soldier, or forward observer, follows a concise communication protocol to identify himself, describe the mission type, describe the target and location, describe the method of engagement, adjust fire as necessary, and conclude with a battle damage assessment. There are a broad range of scenario adaptations that can be enacted to augment the difficulty of a CFF training scenario, such as introduction of obstacles, adjustments to mission type, modifications to enemy behaviors, modifications to weather and time of day, adjustments to type of target and location, and changes to artillery battery response.

Dynamic scenario adaptation involves enacting a series of decisions about how to orchestrate training events at run-time. In DEEPGEN, the full range of possible adaptations is defined in a *Scenario Adaptation Library,* which determines what types of scenario events can be adapted, how they can be adapted, and when they can be adapted. In RL terminology, these correspond to the actions in a Markov decision process, which are enacted at run-time to produce a training experience that meets instructor-specified objectives. The state representation includes both the state of the learner and the history of scenario events to date. Reward is defined in terms of the unfolding scenario's alignment with target instructional objectives. RL provides a systematic process for exploring alternate approaches to dynamic scenario adaptation, gradually improving over time as more trainees interact with the scenario generator.

Figure 2. Virtual Battlespace 3 simulation platform.

To investigate RL-based scenario generation in the domain of CFF training, we utilized example scenarios from Virtual Battlespace 3 (VBS3). Developed by Bohemia Interactive Simulations, VBS3 is a 3D simulation platform that is widely used by the U.S. Army for a range of training purposes, including IED training, surveillance systems, land navigation, route clearance, convoy training, and many other tasks. In this work, we utilize the VBS2Fires plug-in, a third-party tool created by SimCentric Technologies that provides a GUI interface and ballistics simulation engine for training CFF in VBS3 (Figure 2). Automatic scenario generation, which is performed by modifying example VBS3 scenarios provided as input to DEEPGEN, is realized in VBS3 by implementing an automated, or semi-automated, compilation process that produces executable VBS mission files.

As a preliminary investigation of RL-based scenario generation, we implemented a prototype scenario generator that utilizes a multi-armed bandit formalism for inducing policies to generate initial conditions of CFF training scenarios (Rowe et al., 2018). Multi-armed bandits are closely related to RL, but they do not account for the stochastic effects of actions on the state of the task environment, making them a natural starting point for technical demonstration purposes. We utilized a multi-armed bandit approach to induce policies for selecting the weather, time of day, and target movement characteristics in an example CFF training scenario. We considered three possible values of weather (clear, cloudy, rain), 3 possible values for time of day (day, dusk, night), and two possible values of target movement (stationary, moving). To train the multi-armed bandit policies, synthetic data generated from a simple probabilistic simulated learner model was utilized. We ran 50,000 trials of an 18-armed bandit using the UCB1 algorithm to manage exploitation/exploration of different scenario adaptations. Results showed that the scenario generator converged on a stable ranking of alternative training scenarios over time, recommending "easier" scenarios for low competency simulated learners and "harder" scenarios for high competency simulated learners. Although the analysis did not involve modeling sequential decisions about scenario adaptations, it did demonstrate the potential for solving automated scenario generation tasks using RL techniques (Rowe et al., 2018).

Initial Findings

Presentation of Many Variants for Instructor Usability

The first demonstration resulted in several evolutions of presentation for training content like varying scenarios. The underlying novelty search algorithm can vary training in up to eight dimensions per learning objective (not just one, support versus challenge), as described in Dunne, Sivo, and Jones (2015). The dimensions are hypothesized to be domain-independent, so an early idea was to present the dimensions of variation to end users, explaining exactly how each variant differed from the others. Methods that were attempted included arraying many dimensions into visual rows, and selecting two or three dimensions for display in (x,y) space similar to Figure 1. Dimensions could be selected by their range or variability or combined for display via projection. These early attempts were visually complex and offered details that instructors probably do not need to consider.

A second prototype was created (Figure 3**Error! Reference source not found.**). The key features of this prototype include summarizing all dimensions of variation into just three bins per learning objective (easy, medium, and hard), as well as placing a "top five" scenario list front and center, rather than showing every available variant. The list priority was defined by data captured during usage, and was again designed to be domain-general. Usage data included number of times a variant had been used, average duration, and average pass rate. The parameters were intended to work for multiple instructional domains and forms of instructional delivery, and overall to capture institutional knowledge of which variants were more useful. Each variant also received a random, two-word mnemonic to let instructors remember and search for familiar variations.

One result from engaging with a SME is feedback that will lead to a third iteration of usable interface. Military instruction in many domains is described by a three-dimensional matrix. The three dimensions are similarly defined in different domains (Sanders & Dargue, 2012): training complexity, mission, and mission conditions for a command staff trainer; weapons platform, target array, and environment for a gunnery trainer; or task complexity, threat level, and environmental factors for an SUAS trainer. The fine-grained dimensions of variation were initially defined to enhance the classic three dimensions, but an important lesson is that the instructors and instructional designers are typically accustomed to working within the three similar dimensions. Therefore, a third prototype should translate the many underlying variations back into just three dimensions, to provide visual shorthand and explanation of how each scenario varies.

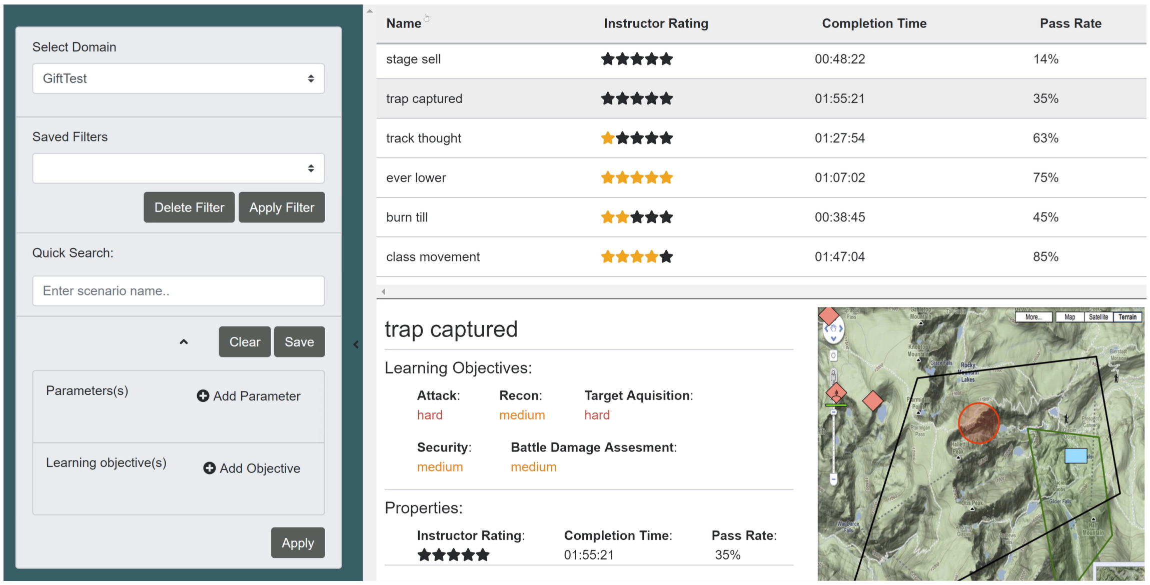

Figure 3: Filtering and ranking variants based on usage parameters and challenge for learning objectives.

Developing Instructor Tools for Dynamic Training Scenario Adaption

Building upon our proof-of-concept demonstration of a multi-armed bandit approach to automated scenario generation, the second demonstration proceeded by investigating two complementary directions: (1) expanding the Scenario Adaptation Library to broaden the space of generatable scenarios while preserving military relevance for real-world CFF training use cases, and (2) designing and developing a prototype DEEPGEN instructor tool for integrating dynamic scenario adaptation capabilities within adaptive instructional systems, such as GIFT. To ensure project alignment with the requirements of U.S. Army training for CFF, we engaged in iterative cycles of feedback with an Army SME bringing extensive experience in CFF training and adaptive training systems.

First, the Scenario Adaptation Library was expanded to incorporate 13 additional adaptable event sequences beyond the 3 utilized in the multi-armed bandit demonstration. This yielded 16 possible dimensions for dynamic scenario adaptation, each with 2-5 possible levels, corresponding to more than 1,000,000 possible variations that could be generated from a single example training scenario. Several adaptable event sequences could be generalized across multiple example scenarios, such as target type (e.g., wheeled vehicle, tank, bunker) and target behavior (e.g., stationary, on patrol), whereas other adaptable event sequences were tied to particular example scenarios, such as the counter-attack behavior of a specific enemy unit. After developing the expanded Scenario Adaptation Library, we obtained SME feedback on how well the expanded set of adaptable event sequences covered the range of useful CFF training scenarios across difficulty levels and instructional objectives. Further, the SME provided input on scenario elements that lacked realism or required refinement for relevance to Army training purposes. For example, SME input addressed issues such as how terrain and target location can impact scenario difficulty, and common target types of call-for-fire missions.

Next, we began to devise user interface mockups for a DeepGen instructor tool to configure automated scenario generation functionalities in adaptive training systems. The tool was designed for use by military instructors and training content developers, and it was envisioned to support eventual integration with GIFT. Three complementary modes of automated scenario generation were targeted as use cases: (1) offline scenario generation, (2) online scenario generation, and (3) run-time scenario generation. In offline scenario generation, an instructor and/or developer utilizes a tool to produce scenarios *prior to a training exercise*. This has labor-saving benefits, and it also expands the range of possible scenarios that can be created for training. However, offline scenario generation does not support scenario personalization, as it lacks access to an explicit learner model that captures a trainee's state (e.g., knowledge, skills, abilities) or trait information (e.g., prior knowledge, goal orientation). In contrast, online scenario generation produces tailored scenarios *just-in-time during training* by consulting a learner model that reflects the trainee's prior performance and competency levels. Online scenario generation is analogous to the outer loop of an intelligent tutoring system, where pedagogical decisions about problem selection are based upon a student model that is maintained by the system (VanLehn, 2006). The third mode of automated scenario generation, run-time scenario generation, takes this process one step further, enacting scenario adaptation *while the trainee is completing a training exercise*. This is analogous to the inner loop of an intelligent tutoring system, where pedagogical support is delivered to guide the learner through the completion of a problem-solving scenario (VanLehn, 2006). In run-time scenario generation, scenarios events are dynamically tailored based upon the learner's performance within the scenario thus far. We distinguish between these three modes because they have distinct implications for the design of instructor-facing tools to support automated scenario-generation use cases, as well as the underlying algorithmic techniques used to implement them.

The purpose of the DeepGen instructor tool is to provide instructors and developers with the ability to specify what types of training scenarios they seek to be generated, as well as preview generated scenarios prior to execution in VBS3. The workflow for using the tool is as follows. The first step is to select a training domain.[1] Next, the user (optionally) uploads example VBS3 training missions, expanding the set of base scenarios for RL-based scenario adaptation. Several example VBS3 missions are provided by default. The user can also upload configuration files that specify the current Scenario Adaptation Library and Performance Assessment Logic for the training domain, which are prerequisites for effective RL-based scenario generation.

After completing these configuration steps, the user selects criteria to guide automated scenario generation through a menu-based interface (Figure 4). Initially, two types of scenario generation criteria are offered: Target Skills and Scenario Difficulty. A range of target skills for CFF training are enumerated, including different methods for specifying the coordinates of a target, performing effective adjustments to fire, and providing an accurate battle damage assessment. Difficulty levels include easy, medium, and hard, and these designations are determined based upon input from SMEs. The user can also toggle into advanced mode, which provides more granular control over scenario generation. In advanced mode, the user can provide input on the types of artillery utilized, method of engagement, types of terrain, visibility conditions, and provision of hints and feedback in the CFF training scenario.

In offline scenario generation, the user next presses a "generate" button, having provided a set of input criteria, to obtain a ranked list of automatically generated CFF training scenarios. For each scenario, a card-like view presents summary information about the mission, including usage data, target skills, key scenario properties, and a score metric derived from the expected reward associated with that scenario in reinforcement learning. The scenarios are ranked according to the score metric, which captures the observed

---

[1] Currently, the only supported domain for automated scenario generation is CFF training. However, the overall approach to scenario generation that is embodied by DeepGen is anticipated to generalize to additional training domains.

effectiveness of the scenario in meeting the user-specified criteria. These scores are updated over time as learners interact with DeepGen, refining the system's model of scenario effectiveness based upon the results of reinforcement learning. When a user clicks on a scenario card, he/she can view a more detailed summary of the mission, which is presented in a standard warning order format, providing information about the situation, mission, execution, task organization, and commanders intent. Generated scenarios can be saved to a library of VBS3 missions for subsequent execution during training.

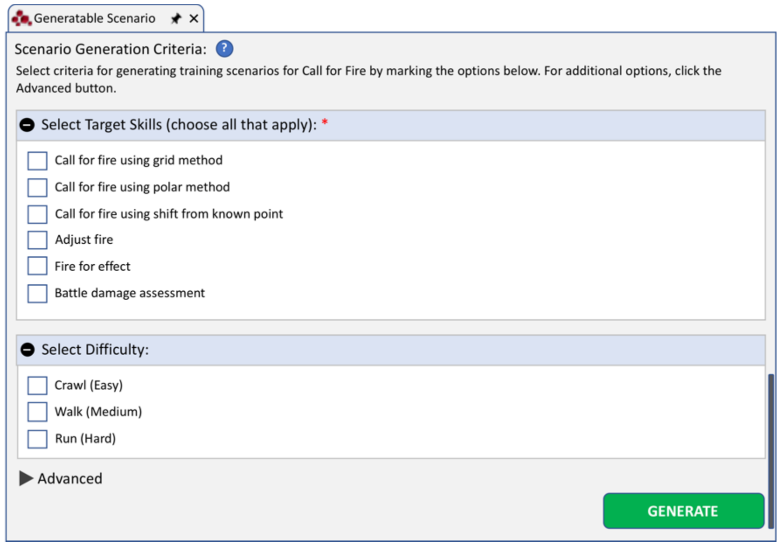

Figure 4. Mockup of DeepGen instructor tool showing default configuration options for automated CFF scenario generation.

The workflow described above is contrasted with anticipated workflows for online scenario generation and run-time scenario generation, respectively. In these latter modes, a "generate" button is unnecessary, because scenarios are generated dynamically during training by tailoring within-scenario events to the individual characteristics of learners. Online scenario generation can be conceptualized as a pedagogical event within a broader instructional sequence, which could also include embedded assessments, direct instruction, and practice with manually-crafted scenarios, for example. Based upon this observation, we have begun to create UI mockups of the DeepGen instructor tool that envision automated scenario generation as a course object within the GIFT Course Creator. Devising instructor-facing tools for automated scenario generation that are compatible with existing lesson builders, such as the GIFT Course Creator, will be critical for bringing online scenario-generation use cases into reality.

It should be added that in online and run-time scenario generation, instructors will almost certainly seek the ability to preview how scenario generation systems will operate for different types of learners. Factors such as *transparency* and *explainability* are critical to establishing the trust necessary for human instructors to adopt AI-based technologies, such as automated scenario generation, in their training workflows (Sinha & Swearingen, 2002). Devising methods and tools for visualizing how dynamic scenario adaptation features operate within DeepGen is the subject of continued work.

To guide the iterative design and development of the DeepGen instructor tool, we have engaged SMEs in several rounds of feedback on its user interface design. SME feedback has been instrumental in refining the terminology and criteria featured within the instructor tool, including the CFF skills that are targeted, the types of artillery that are supported, and types of terrain that can be utilized. Specifically, SME feedback led to the addition of configuration options for the method of engagement in CFF (e.g., type of adjustment, trajectory, ammunition, danger close), advised adding terrain options that contain tree cover to enhance scenario difficulty, and suggested refining criteria for time pressure to distinguish between diegetic time pressure (e.g., enemy forces launching an attack) and non-diegetic time pressure (e.g., a time limit).

Discussion

The first demonstration highlights a need in the current state of practice to usefully present many training variations—filtering, finding, and describing the variants presented in a way that aligns with what instructors want to know. Good alignment will allow instructors to start using a sophisticated training system, understand its recommendations, and accept or change them to optimize learning. Poor alignment will increase the barrier to entry for new users and reduce the job effectiveness of any instructors who do not give up on using the training entirely. Because a sufficiently sophisticated computational system may consider hundreds or thousands of factors in recommending an adaptive choice of scenarios or interventions, there is a challenge to identify how these factors, across training domains, should be presented to align with end user needs.

The findings indicate that instructors share a common vocabulary and mental model which are found to appear in multiple sources and training domains. The constructs of task complexity, level of risk or urgency, and environmental conditions should form a template by which more fine-grained variation is understood. Once these three dimensions are accepted as a basis for describing training variants, the dimensions can be applied to different training domains and can be refined by additional measures. If a computational system requires or offers many more dimensions of variation, they may be initially hidden within a summary or rollup metric and available for drill-down when advanced users request more information.

Institutional knowledge, or information about training that is accumulated over time and through repeated use, is likely to provide a secondary or optional entry into the training choices available. In the military use case, achieving the mandated training is paramount and therefore the primary data presentation should support easily identifying mandatory training within a sequence. The three-dimensional model is likely to help filter and find training content for this use case, with qualities such as duration and pass rate available as secondary filters after these.

The findings of the second demonstration illustrate how the intrinsic combinatorics of dynamic scenario adaptation yield a vast range of generatable scenarios for even simple domains, such as CFF training. By integrating additional example scenarios, or devising broader missions that embed CFF within them, the yields of automated scenario generation can be increased further. This observation underscores the importance of devising methods to provide instructors and developers with control over the operation of automated scenario generators. In the DEEPGEN instructor tool, we provide users with general criteria, such as target skills and scenario difficulty, as well as granular criteria, such as domain-specific methods of engagement and adversary behaviors, to guide scenario generation of CFF training missions. Similar to the first demonstration, we find that it is critical to work closely with SMEs to guide the formulation and presentation of control methods to ensure that instructor-facing tools are understandable, usable, and useful. Further, the second demonstration highlights the promise of devising empirically based evaluations of scenario quality that can be leveraged to rank and assess candidate training missions. In RL-based scenario generation, this evaluation mechanism is implemented algorithmically in the form of a reward model that is induced from data on learner interactions with candidate scenarios as well as their performance and training outcomes.

This work also highlights the different ways that automated scenario generation can be integrated into real-world training workflows. Automated scenario generation can be utilized as a labor-saving tool, reducing the costs of developing training scenarios through offline scenario generation processes. Additionally, automated scenario generation can be utilized online, and at run-time, to enhance adaptive training capabilities through dynamic personalization of scenarios in line with the individual characteristics of learners. These complementary modes have significant implications for the design of instructor tools for controlling the operation of scenario generators. In offline scenario generation, an instructor is likely to peruse candidate scenarios, save them to a library of training materials, and deploy them to learners. In online scenario generation, as well as run-time scenario generation, an instructor is likely to seek understanding of how dynamic scenario adaptation will shape learner experiences during a training exercise based upon learners' individual states and traits. Automated scenario generation creates the need for supporting transparency and explainability within instructor tools, which will be critical to establishing the trust necessary for instructors to adopt automated scenario generation technologies in the classroom.

Recommendations and Future Research

Research on automated scenario generation is still in its nascent stages, and there are several promising directions for future research. For the two projects described in this chapter, continued engagement with SMEs, including instructors and training content developers, will be essential for refining the scenario generation tools to support real-world training use cases. In addition, it will be important to investigate how these tools, and their underlying novelty search and RL-based approaches to scenario generation, respectively, generalize to additional training domains. Third, conducting evaluation studies to investigate the prospective labor-saving benefits of automated scenario generation, as well as the training effectiveness of created scenarios, will be critical to develop an evidence base for the benefits of automated scenario generation in adaptive training systems for military domains.

More broadly, there are myriad open questions about automatic scenario generation that require further attention. First, it will be important to investigate the relative strengths and weaknesses of alternative computational frameworks that have emerged in recent years, such as generative adversarial networks, for solving automated scenario generation tasks. This calls for methodological progress in the evaluation of automated scenario generation systems, including identification of appropriate instruments, metrics, and research designs that reveal the effectiveness of alternative scenario generation approaches. Second, investigating mixed-initiative systems that enable human instructors and content developers to co-create training scenarios in coordination with automated scenario generation systems has significant potential. Finally, devising examples of how to integrate automated scenario generation functionalities with existing tools for constructing adaptive training systems, such as the GIFT Course Creator, will be critical for taking scenario generation out of the laboratory and into real-world classrooms.

Conclusions and Recommendations for GIFT

The two studies presented in this chapter illustrate recent advances in automated training scenario generation that hold significant promise for real-world training applications. Automated scenario variation before training, and dynamic scenario adaptation during training, are well positioned to help reduce the human effort and cost associated with generating tailored, effective instruction and assessment. Addressing practical considerations in effective deployment and use of such research will help to enrich the training capabilities of adaptive instructional systems such as GIFT, as well enable the creation of adaptive training systems that continually improve in effectiveness and utility over time. The inputs needed for each of these systems are the instructional objectives, example scenarios that target them, and some amount of student data about experiences with the scenarios. These items are required to generate the scenarios and need to be represented through metadata tags, descriptors, folder structures, or equivalent. For output, the system must either have a link to (1) an instructor interface to select student scenarios, (2) a system interface to automatically assign the scenarios, or (3) both. Adaptive instructional systems must have a way of describing existing content to algorithmic content generators, as well as links to where this content can be placed after its creation.

References

Dunne, R., Sivo, S.A., & Jones, N. (2015). Validating scenario-based training sequencing: The scenario complexity tool. *Proceedings of the the Interservice/Industry Training, Simulation and Education Conference* (No. 15102).

Folsom-Kovarik, J.T., & Brawner, K. (2018). Automating variation in training content for domain-general pedagogical tailoring. *Proceedings of the Sixth Annual GIFT User Symposium* (pp. 75-86). Orlando, FL: U.S. Army Research Laboratory.

Gomes, J., Urbano, P., & Christensen, A. L. (2012). Progressive minimal criteria novelty search. *Proceedings of the Ibero-American Conference on Artificial Intelligence* (pp. 281-290). Springer, Berlin, Heidelberg.

Lehman, J., & Stanley, K.O. (2008). *Exploiting open-endedness to solve problems through the search for novelty. Proceedings of the Eleventh International Conference on Artificial Life* (pp. 329-336). Cambridge, MA: MIT Press.

Lehman, J., & Stanley, K.O. (2011). Novelty search and the problem with objectives. *Genetic Programming Theory and Practice IX* (pp. 37-56). New York, NY: Springer.

Liapis, A., Yannakakis, G.N., & Togelius, J. (2015). Constrained novelty search: A study on game content generation. *Evolutionary Computation, 23*(1), 101-129.

Rafferty, A. N., Jansen, R. A., & Griffiths, T. L. (2016) Using inverse planning for personalized feedback. *Proceedings of the Ninth International Conference on Educational Data Mining* (pp. 472-477).

Rowe, J., & Lester, J. (2015). Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. *Proceedings of the Seventeenth International Conference on Artificial Intelligence in Education* (pp. 419-428). New York, NY: Springer.

Rowe, R., Smith, A., Pokorny, B., Mott, B., & Lester, J. (2018). Toward automated scenario generation with deep reinforcement learning in GIFT. *Proceedings of the Sixth Annual GIFT User Symposium* (pp. 65-74). Orlando, FL: U.S. Army Research Laboratory.

Sanders, J., & Dargue, B. (2012). *A design whose time has finally arrived: Use of adaptive training technology for staff training*. Presented at the 23rd International Training Equipment Conference (ITEC), London, UK.

Sawyer, R., Rowe, J., & Lester, J. (2017). Balancing learning and engagement in game-based learning environments with multi-objective reinforcement learning. *Proceedings of the Eighteenth International Conference on Artificial Intelligence in Education* (pp. 323-334). New York, NY: Springer.

Shen, S., Mostafavi, B., Barnes, T., & Chi, M. (2018). Exploring induced pedagogical strategies through a Markov decision process framework: Lessons learned. *Journal of Educational Data Mining, 10*(3), 27-68.

Sinha, R., & Swearingen, K. (2002). The role of transparency in recommender systems. *CHI'02 Extended Abstracts on Human Factors in Computing Systems* (pp. 830-831). New York, NY: ACM.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction (Second Edition).* MIT Press.

Wang, P., Rowe, J., Min, W., Mott, B., & Lester, J. (2018). High-fidelity simulated players for interactive narrative planning. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, (pp. 3884-3890).

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, *16*(3), 227-265.