Automating Variation in Training Content for Domain-general Pedagogical Tailoring

J. T. Folsom-Kovarik[1], Keith Brawner[2]
Soar Technology, Inc.[1], U. S. Army Research Laboratory[2]

INTRODUCTION

The Generalized Intelligent Framework for Tutoring (GIFT) (Sottilare, Brawner, Goldberg, & Holden, 2012) is able to tailor training content selection and presentation in order to give individual learners the support or challenge they need (Goldberg, Sottilare, Brawner, & Holden, 2011). However, the effectiveness of tailoring is always limited by the choices available. Automated scenario generation (ASG) offers promise to create many more variations on training content than human experts can create alone. A proof of concept ASG implementation is being researched and developed. The ASG can create variants on training scenarios that encompass multiple types and parts of scenario content to include simulation events and narrative, entity location and size, and feedback or framing text. Combining several such variants will let GIFT simultaneously support and challenge different learning objectives in one training scenario.

A key insight in the work is that generated variants are labeled in a domain-general way according to their predicted impact across learning objectives. As a result, all the details of variations are expressed to GIFT in a manner easily processed by general pedagogical algorithms. Furthermore, the domain-general dimensions of support and challenge for each learning objective define a space within which the ASG components should search. That is, ASG does not simply find random variations on the scenario factors it can control directly, such as entity location and size. Apparent variation in those factors could easily turn out to be different only at a surface level – not different at a level that adds new kinds of support or challenge to the library of choices available to GIFT pedagogical tailoring. A key portion of this work is not the simple generation of many superficially different scenarios, but the generation of scenarios which are sufficiently different from one another while still being pedagogically valid.

Two steps enable ASG to create variants that are valuable for GIFT to tailor learning content. First, ASG must perform search in a space where movements are possible to directly control, such as entity size and location. Second, ASG must translate the variants it made in that space into the space defined by multidimensional impact on learning support and challenge. The first step is carried out by a form of evolutionary algorithm called novelty search. The second step is carried out by defining a cognitive science-based understanding of learning factors and creating domain-specific rules that translate expert knowledge into the terms of the generalized framework.

In the initial days of research and development, ASG is being prototyped in a specific training system for decision-making in the employment of small unmanned air systems (SUAS). The training system contains expert-authored content such as maps with geographical features, tactical objectives and constraints, friendly and hostile entities; text briefing materials and initial conditions; and learner decision prompts and feedback. In the current state of research, an example is presented using map variation. The example illustrates how novelty search programmatically creates candidate arrangements of elements on the training scenario map, then labels each one according to its predicted impact on learning. Specifically those map variations that change the scenario challenge level are stored in a library for the GIFT pedagogical module to select, using the domain-general labels on each variant. As research and development continues, more elements of training such as text content will also be varied using ASG. The result of having many variants is that GIFT may individualize the training experience in order to simultaneously support one learning objective and challenge another, according to learners' needs.

BACKGROUND AND RELATED WORK

ASG has combined relevance across multiple research fields. This section discusses the current state of the art in (1) evolving scenario content, (2) novelty search as an approach to addressing issues of traditional evolutionary algorithms, and (3) computational accounting of pedagogical impact.

First, there have been several successes in procedurally generated content for games or training scenarios. Evolutionary algorithms have generated content such as scenario terrain, behaviors, events, and narrative (Luo, Yin, Cai, Zhong, & Lees, 2016; Stanley, Bryant, & Miikkulainen, 2005; Zook, Lee-Urban, Riedl, Holden, Sottilare et al., 2012). Evolution is well-suited to spaces where there are too many possible variations to explore them all or randomly choose variants to evaluate. In generating training content, effective search is needed because it may not be easy to predict how the changes that are easy to make, such as terrain or unit locations, will affect the desired outcomes, which are to change how instruction works for learners. As a concrete example, changing the position of an enemy unit from “left of the hill” to “right of the hill” may have no training effect for a ground-based tactical movement scenario, but significant training effect for an indirect fire scenario, where accounting for wind velocity and smoke effects is a training objective.

Second, evolutionary algorithms as a class suffer from certain shortcomings that are broad and practical in their importance. Evolution can require days to complete, or may demand high-performance server clusters. A key reason is that evolution typically needs to be carefully tuned to avoid premature convergence on one local optimum, finding the same variants repeatedly instead of new ones. However, a recent advance in evolutionary algorithms suggests that an entirely different approach both yields better results and reduces required computer resources. This approach is novelty search (Lehman & Stanley, 2008, 2011). Novelty search replaces evolution toward a higher fitness with evolution toward increasingly different individuals. Novelty search has been used with success to evolve content similar to training scenarios, such as game levels (Liapis, Yannakakis, & Togelius, 2015).

Third, there is the question of letting novelty search predict the training impact of generated variants. Instructional designers, educators, and cognitive psychologists are among those who have created frameworks for predicting the effect of training scenarios, interventions, and other content on individual learners and in different training contexts (Campbell, Ford, Campbell, & Quinkert, 1998). Two factors that have been recently studied are scenario helpfulness and complexity. Helpfulness describes the *explicit* interventions that can be part of training, such as help messages, hints, or formative feedback—are they specific or broad, immediate or delayed, and so on (Shute, 2008). Complexity gives a good complement by measuring *implicit* interventions which vary scenario content in order to support or challenge learners, such as tailoring the number of enemies or the amount of time remaining to carry out a task (Dunne, Sivo, & Jones, 2015). Measuring different dimensions or categories of helpfulness and complexity has driven tailoring in past work, but has been manually defined for each variant (Folsom-Kovarik, Newton, Haley, & Wray, 2014; Folsom-Kovarik, Sukthankar, & Schatz, 2013; Graffeo, Benoit, Wray, & Folsom-Kovarik, 2015).

In summary, evolutionary approaches may be able to generate meaningful scenarios from the infitine set of possibilities, to do so quickly when using a metric such as novelty search, and objectively measure instructional relevance. The natural divisions of the “genotype” and “phenotype” space within an evolutionary approach lend themselves to representing the literal scenario content (genotype) and its learning impacts (phenotype). An evolutionary content generation method that would also let end users such as instructors and subject-matter experts understand and control the content evaluation in an objective manner would help to improve usability and user acceptance of the approach (Folsom-Kovarik, Wray, & Hamel, 2013).

NOVELTY SEARCH AND APPLICATION TO ASG

Evolutionary algorithms are appropriate methods to search when a space is too high-dimensional, unevenly gradiated, or otherwise inappropriate for simpler enumeration or gradiant descent methods. Evolution typically maintains a notional population of points in the search space which are evaluated to find their fitness for the purpose at hand. The points in the population are then combined and varied with operators that aim to increase fitness of the next generation and remove points that have lower fitness. Novelty search addresses some limitations of evolutionary algorithms. Instead of working to increase fitness, the aim is to increase novelty and explore points that are as different as possible from what has been seen before. In this way, novelty search attempts to remove premature convergence concerns typical in evolutionary algorithms and produce many variants that can be filtered for fitness to a specific need. This is specifically an advantage in the training domain space where differences among the scenarios is an expliticly stated goal.

This section describes the current state of the novelty search algorithm under development. Novelty search efficiently finds variants that are new in a domain-general sense. That is, the variants provide a different manner of support or challenge than any variant already available. As novel variants are created offline, they can be stored for human review and access by GIFT. The novelty search is an “anytime” process meaning that it can provide results immediately or continue to improve the results as more compute time is available when not actively training. GIFT is then able to select in real time during training between variants using its existing, domain-general Pedagogical Module. Instructors will be able to see what variants are available and identify any gaps that still need to be evolved. This allows for both the generation of content that the instructor can approve and for the further development of training exercises for students if the amount of approved content is exhausted.

The current implementation of novelty search is built on the open-source library Distributed Evolutionary Algorithms in Python (DEAP) (Fortin, Rainville, Gardner, Parizeau, & Gagné, 2012). DEAP offers a combination of fast prototyping now with fast computation and variant generation in future deployment. Like other evolutionary algorithms, novelty search depends on effective design of (1) genotype representation, (2) genetic operators, and (3) fitness function or in this case novelty evaluation function. In the current state of research, these are domain-specific. However, future research may be able to identify opportunities for reuse across broad domains such as the spaces of all images or all text documents.

First, ASG differentiates genotypes from phenotypes in this way. Genotypes are those objects, such as strings of digits, that can be easily changed during evolution, while phenotypes are the objects that can be evaluated for their novelty. The phenotype is the scenario variant itself, that which the learner experiences during training and which the instructor must agree provides appropriate support and challenge. As a result of this difference, evolution is not needed when it is possible to jump straight to the desired phenotype. Instead, the separation of genotype and phenotype is necessary because the phenotype can be measured on dimensions that matter to instruction, like complexity and helpfulness, but the genotype cannot be measured until it is transformed into a phenotype. Conversely, at the genotype level there are a set of changes which are easy to apply to generate new variants, but it turns out to be difficult to make changes at will in the training experience phenotype, because human creativity would be required.

The genetic representation currently used throughout the rest of this work in ASG is a direct representation. More complex representations such as neural networks or hypercubes (Kocmánek, 2015) have also been used in novelty search but were deemed unnecessary at this stage of development. The genetic representation encodes each element of the evolved training scenario one-to-one in a vector of descriptive values. For example, in order to evolve locations of objects in a two-dimensional space, the genetic representation would describe each object with its type, x-coordinate, y-coordinate, and perhaps scale or rotation values. The current representation describes points, lines, and regions in two-dimensional space, which is hypothesized to be extensible to multiple domains.

Second, genetic operators in ASG are designed to make changes to genotypes. The changes are not guaranteed to produce a better genotype, but they should be designed to build on what has already been evolved and create new genotypes that have a reasonable possibility to be viable. The genetic operators used are element insertion, single-point mutation, and single-point crossover. Element insertion increases the complexity of lines and regions by adding a new point at random to the genotype. Single-point mutation chooses a numeric value uniformly at random and changes it by adding a random perturbation with Gaussian distribution about zero. The crossover operator combines two existing genotypes by choosing a point in the vector and taking all elements to the left of that point from one genotype, all elements to the right of that point from the other. Since direct representations have been well-studied in other evolutionary algorithms, these operators are standard in the field and do not introduce additional domain specificity.

Third, the novelty evaluation function in ASG is the tool that determines when evolution has produced a variant that is new in an interesting way, as opposed to a variant that appears to be new on the surface but does not provide any difference in training support or challenge. The terms "support" and "challenge" are considered to be opposite ends of a single continuous scale for the purposes of this work. The evaluation phase consists of applying domain-specific rules to each phenotype (training variant) in order to find its value on domain-general dimensions. Four domain-specific rules were created as a proof of concept and are described in the ASG Example section below. The domain-general measures that result from these rules describe facts about the training such as complexity of meeting one learning objective or another.

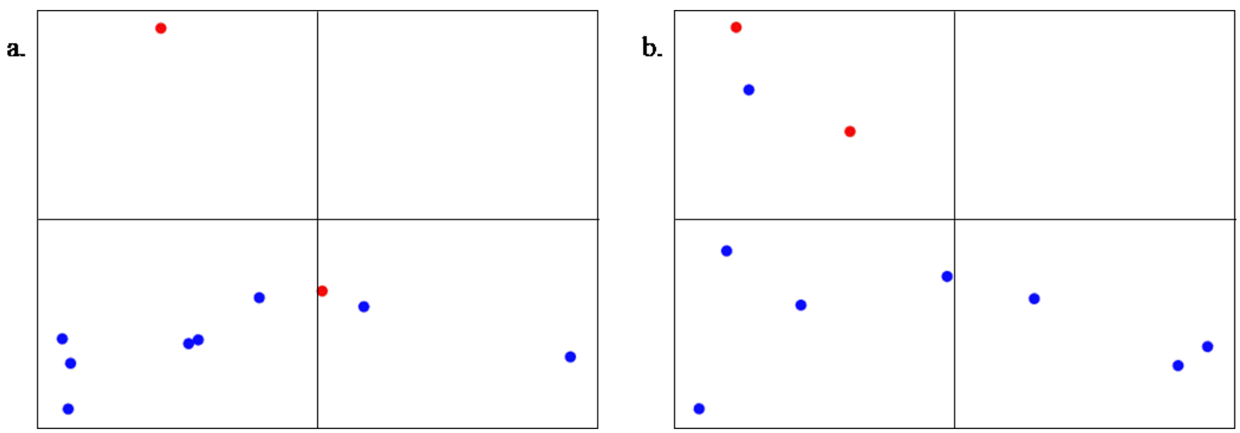

Figure 1: Training complexity (a) in the first generation and (b) after 20 generations. The different point colors and x,y locations (spread) visualize the diversity of training options that the evolved variants offer.

ASG determines which variants are novel in the training challenge sense by clustering variants on the training complexity values, finding the k nearest neighbors (k=2 for efficiency), and selecting the variants which maximize Euclidean distance from their respective nearest neighbors. A hall of fame was maintained to provide persistence across generations of the current maximally novel individuals. In this scheme, different factors that separately affect complexity formed additional dimensions in the complexity measure. Examples were number of distractors or time constraints. As such, evaluation was found to require a scaling step in order to make different dimensions comparable and prevent one dimension from outweighing others.

The outcome of the overall algorithm for novelty search is an increasingly diverse collection of training variants. Figure 1 demonstrates the difference between an initial generation and the variation after 20 generations. The example complexity measures described below are projected into two-dimensional space. The increasing distance between the individuals after evolution indicates that novelty search produced variants which provide GIFT with more choices between noticeably different levels of challenge and complexity.

A DOMAIN-GENERAL REPRESENTATION OF SCENARIO CONTENT

The creation of a domain-general representation of learning impact enables contributions from many instructors, authors, and researchers to work together to increase GIFT tailoring options. A computational representation of factors that impact learning also enables automatically evaluating what is novel in ASG and what will help learners at scenario runtime. GIFT has previously conducted a literature review to support the selection of domain-general factors, including complexity. This proof of concept adds to that review a high degree of precision that breaks down support and challenge into multiple contributing factors that can be separately measured, varied, and objectively compared across learning domains and systems.

Diverse instructional theories suggest categorizing or sequencing learning tasks based on a continuum of complexity. Gagné (1965) organized learning tasks into a broad hierarchy consisting of stimulus recognition at the least complex through application and problem solving as the most complex tasks for any particular identified skill. A similar concept of task complexity has been used in past research with a Dynamic Tailoring System (DTS) that could choose in real time between training variants that were labeled by human experts with scalar complexity values (Wray & Woods, 2013).

In the context of ASG, a single scale for task complexity is not expressive enough to capture the different ways in which the same task can be varied to be more or less complex. Many generated variants are likely to have the same complexity when measured on a single scale – an example might be a hypothetical GIFT question bank that contains a hundred different multiple choice items. Empirically some challenge learners more than others and are answered correctly less often. However, they might all be evaluated as equal in complexity because they all require simple recognition (of the correct choice). In the multiple choice situation, which seems not atypical, two possible approaches could let a computer system differentiate the available variants in advance without human expert labeling. First, GIFT could have an extremely fine-grained hierarchy of sub-concepts. In this case, GIFT could differentiate and sequence the available variants based on hierarchical relationships between the sub-concepts such as prerequisites. This approach is not attempted in ASG. The second approach, which is explored here, is to increase the dimensions by which variants may be described. Complexity itelf must be analyzed in more detail.

Dunne, Cooley, and Gordon (2014) conducted an initial analysis of factors that contribute to learning complexity. These factors included task complexity factors such as number of actions required and number of interdependent actions, as well as learning context factors such as number of possible ways to complete a task and number of distractors. These factors appear in Table 1. On the other hand, Table 1 also introduces a notional definition of helpfulness. As a complement to complexity, helpfulness has not yet been operationalized to provide concrete measures and will be discussed here at an early stage of exploration.

First, Dunne and colleagues suggest theory-based, countable measures that help provide a multidimensional framework for objectively measuring complexity. Complexity increases with each of the factors in Table 1, although possibly nonlinearly. Current work with ASG is working to build rules that predict the impact of scenario variants by counting factors such as the number of cues, actions, and distractors in each variant. Each dimension in the framework is furthermore related to one equation that calculates scenario complexity and has been initially validated with empirical study of a military training sequence in the same citation. The rules that carry out counting the complexity factors are domain-specific, but they result in domain-general measurements. The domain-general measures let the GIFT pedagogical module work without domain-specific knowledge and enable objective comparison across variants.

Table 1: Domain-general dimensions describing challenge and support for each task or learning objective.

Measuring Complexity		Measuring Helpfulness
Number of cues		Attention via perceptual arousal
Number of actions		Attention via inquiry arousal
Number of subtasks across actions		Relevance via previous link
Number of interdependent subtasks		Relevance via needs link
Number of possible paths		Confidence via evaluation link
Number of criteria to satisfy		Confidence via learner control
Number of conflicting paths		Satisfaction via feedback positivity
Number of distractors		Satisfaction via future link

Second, a measure is needed which describes the dimensions on which extrinsic or direct interventions can be measured. Interventions such as hints, help messages, and text documents that deliver remediation vary in their helpfulness with respect to specific learning objectives. As an intuitive example, a help message delivered inside a scenario by a character over the radio can be clear, concise, and on point to provide support, or it can deliberately challenge learners by being ambiguous, wordy, or distracting. A cognitive science basis for enumerating the possible differences in how helpful scenario components are lies in the ARCS model of instructional design (Keller, 1987). This model describes factors of attention, relevance, confidence, and satisfaction that motivate an adult learner to engage with learning content. Unlike the Dunne model, research is still needed to produce an accounting of how a computer system can see factors in this model. One example that moves the ARCS model toward countable dimensions might be a heuristic measure of inquiry arousal from counting the number or frequency of keywords like “how” and “why.”

The combination of complexity and helpfulness is hypothesized to provide ASG with multiple objective measures to describe and differentiate the impact of every variant on different learning objectives. In this way, a domain-general representation of scenario content is hypothesized to increase the opportunities to apply learning theory in GIFT’s automated design and selection of content that is tailored for learners. ASG can augment a hierarchical analysis or a fragmented, parts-to-whole sequencing with recommendations that reflect how adults learn material of increasing complexity in context (Reigeluth & Stein, 1983).

ASG EXAMPLE FOR SCENARIO LAYDOWN

ASG is being developed and evaluated in the context of existing training for proper use of small unmanned air systems (SUASs). The training consists of sequential problem presentations in the context of a narrative supported by mission briefings and maps depicting the area of operations (Figure 2). Learners are also presented with adaptive hints and texts for remediation depending on their performance. The system has been designed to teach nine terminal learning objectives and 48 enabling learning objectives, a huge number of dimensions for evolution to explore if all combinations of learning objectives can eventually be varied in complexity and helpfulness. In the present research and development, a subset of three learning objectives was chosen for initial exploration. The first target for evolution was the mission map. Future work is planned to evolve text-based content such as briefings, pop-up events, and hints or remediation documents.

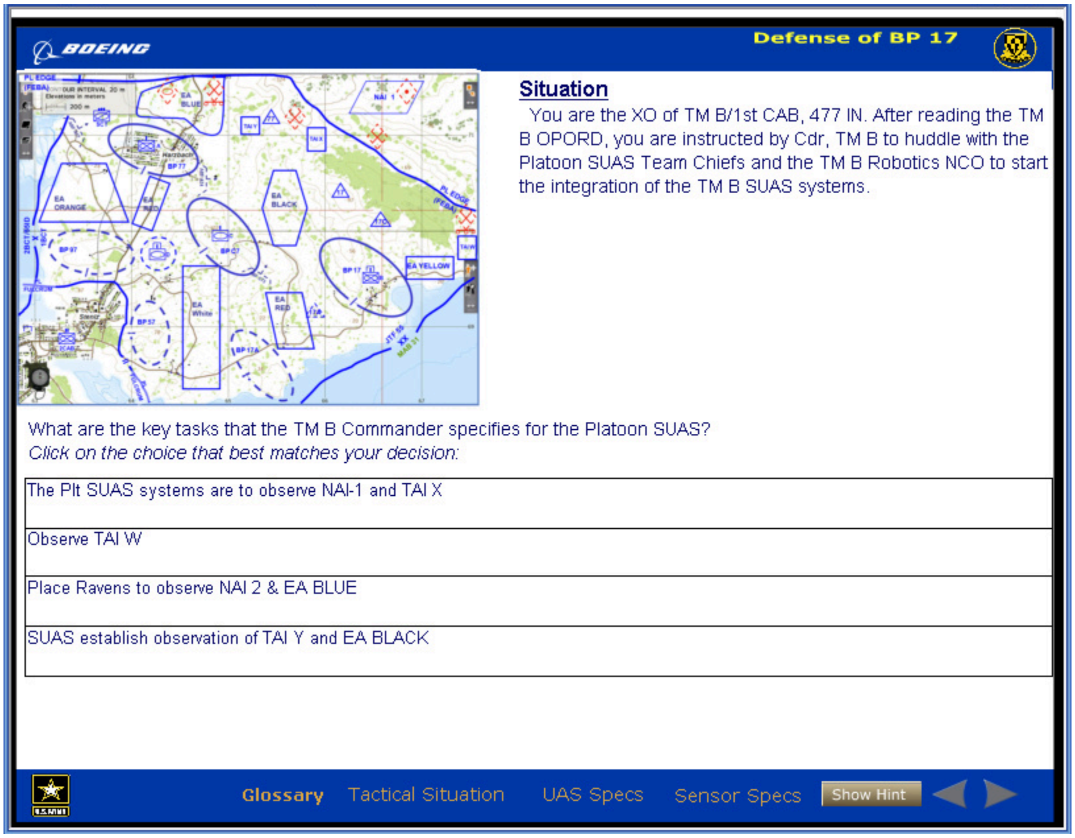

Figure 2: The SUAS training domain for developing and evaluating ASG.

The ASG example seeks to evolve variants on the mission map depicted in the top left of Figure 2. In this example, the elements that can be evolved are shown in Figures 3 and 4. These include the locations of friendly and hostile units, a no-fly zone (red oval), and the shapes of roads, water, and forest terrain features. According to the ASG algorithm described above, these elements could be moved easily on the generated map variants. The next step was to demonstrate how the variants could be measured in training complexity space and selected as being more or less novel from a training impact perspective.

Three training complexity measures were created for the example implementation. The measures were simple rules reflecting three of the learning objectives in the real training system: enemy air defense avoidance, recon and surveillance, and airspace coordination procedures. The simplified rules showed examples of three dimension types: continuous scalar, discontinuous scalar, and categorical. (1) For enemy air defense, one rule was created that stated training complexity increases with proximity to an enemy unit. The enemy was considered to have air defense capabilities that made it difficult to operate when near the enemy. (2) for recon and surveillance, two rules defined one complexity dimension. If an enemy unit was located within a forest region, the complexity of the training increased with the size of the forest. If the enemy was outside a forest, the complexity decreased in proportion to the distance from the enemy to the nearest edge of a forest region. (3) For airspace coordination, the rule was that complexity was high when a no-fly zone lay between the enemy and friendly units, while complexity was low otherwise. In Figures 3 and 4, red dots indicate scenarios with high complexity in this dimension while blue dots have low complexity.

During novelty search, the first generation maps (Figure 3) typically did not even contain both a friendly and a hostile unit. This helps explain why they are all rated as similar in the complexity of air defense avoidance (Figure 1 above).

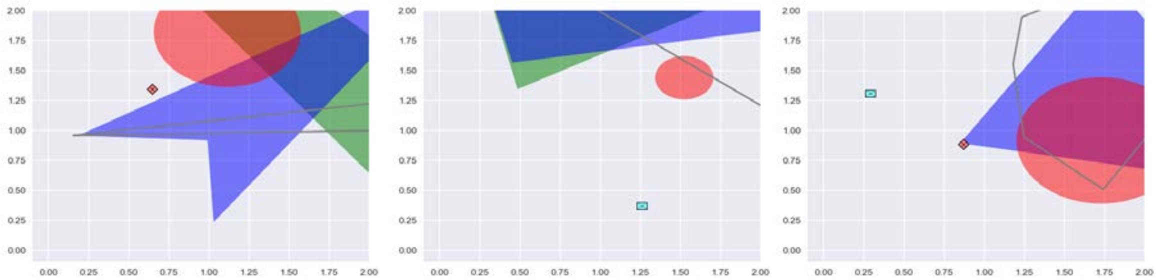

Figure 3: First generation of evolving scenario variants.

The last-generation maps (Figure 4) have evolved a greater frequency of having one friendly and one hostile unit on the map, which probably helped to explore more possible values of air defense avoidance complexity and let ASG provide variants at more places on this scale.

The last-generation maps also display increased complexity of the contours around water and forest regions. This is an example of a difference that is visually very apparent but makes no difference for the purposes of measuring training complexity. The value of novelty search using domain-general measures of the variants is that the simplicity or complexity of the scenarios are evaluated without regard to surface details except where a rule tells ASG that those will change the learning experience.

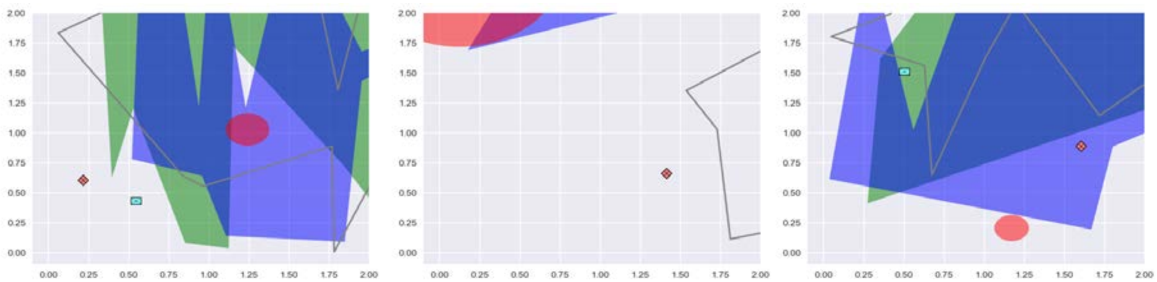

Figure 4: Last generation scenario variants.

In summary, the work of developing an example of ASG in a GIFT-enabled training domain has helped to develop some of the proposals and surface findings in this paper, as well as considerations that will be addressed in ongoing research and development. The initial novelty search examples presented here used only a small fraction of the potential dimensions that could be measured to describe the SUAS training. As a result, there is great potential for novelty search to create a large library of scenario variants that offer GIFT any desired combination of support and challenge for delivering tailored training.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Presented above are the first steps in the investigation of Automated Scenario Generation. This research divides the problem up into three problems – downselecting from an infitine number of possible scenarios, doing so in reasonable time, and making scenario variants which are instructionally relevant. The proposed approach uses genetic algorithms, with a novelty search fitness function and a domain-general representation of scenario content, to enable variant selection without a specialized pedagogical module and to present to instructors and students. Generally, the above research is incomplete – it did not include an analysis on whether the generalized scenarios are useful to SMEs or what dimensions of changes are needed or desired. This work is yet to be performed.

This work, thus far, is able to make use of the existing GIFT logic, structure, and modules. The scenario selection logic within the pedagogical module should be written in a general enough manner so as to be able to be applied to a large number of generated scenarios. Performance assessment within the generated variants is also needed. Fundamentally, GIFT will have to provide the same tutoring to the new scenarios as it does to the old – the functionality is built into GIFT and the DKF structure. Integration at the end of the project may be as simple as a pointer to the optimal fit in a library of generated variants when the student reaches the appropriate experience at runtime. In the ideal case, performance assessment could be dynamic and follow rules drawn from or similar to the novelty evaluation rules, in the same manner as the DKF currently allows for pointers to external assessment engines.

Next steps in the near term will be to replace the illustrative ASG example presented in this paper with more realistic rules for complexity. Instructor and SME interviews are planned to determine which learning objectives and aspects of the real training system are likely to be most impactful to vary. Understanding what dimensions human expert instructors actually want and need to vary will inform a more comprehensive set of rules that will help test and improve the efficiency and effectiveness of the ASG approach.

In regards to future work, scenarios represent the most complex piece of content represented within GIFT. The example ASG could easily be extended to other two-dimensional content like images or VBS terrain. Simpler pieces of content, such as prompted, hints, feedback strings, webpages and other items are also shown to the user in GIFT training but are not procedurally generated. Technology to procedurally generate these types of items may have to be implemented differently, as these items may rely on text or image processing techniques rather than modification and generation techniques, but there may be another class of ASG representations and operators that is effective for many types of text content.

REFERENCES

Campbell, C.H., Ford, L.A., Campbell, R.C., & Quinkert, K.A. (1998). *A Procedure for Development of Structured Vignette Training Exercises for Small Groups* Alexandria, VA: Human Resources Research Organization.

Dunne, R., Cooley, T., & Gordon, S. (2014). *Proficiency Evaluation and Cost-Avoidance Proof of Concept M1A1 Study Results.* Paper presented at the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), Orlando, FL.

Dunne, R., Sivo, S.A., & Jones, N. (2015). *Validating scenario-based training sequencing: The scenario complexity tool.* Paper presented at the Interservice/ Industry Training, Simulation and Education Conference (I/ITSEC), Orlando, FL.

Folsom-Kovarik, J.T., Newton, C., Haley, J., & Wray, R.E. (2014). *Modeling Proficiency in a Tailored, Situated Training Environment*. Paper presented at the 23rd Conference on Behavior Representation in Modeling and Simulation (BRIMS), Washington, DC.

Folsom-Kovarik, J.T., Sukthankar, G., & Schatz, S. (2013). Tractable POMDP representations for intelligent tutoring systems. *ACM Transactions on Intelligent Systems and Technology (TIST), 4*(2), 29.

Folsom-Kovarik, J.T., Wray, R.E., & Hamel, L. (2013, July 9-13). *Adaptive assessment in an instructor-mediated system.* Paper presented at the 16th International Conference on Artificial Intelligence in Education (AIED), Memphis, TN.

Fortin, F.-A., Rainville, F.-M.D., Gardner, M.-A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research, 13*(Jul), 2171-2175.

Gagné, R.M. (1965). *Conditions of Learning*. New York, NY: Holt, Rinehart, and Winston.

Goldberg, B., Sottilare, R., Brawner, K., & Holden, H.K. (2011). *Predicting learner engagement during well-defined and ill-defined computer-based intercultural interactions*. Paper presented at the HUMAINE Association on Affective Computing and Intelligent Interaction, Memphis, TN.

Graffeo, C., Benoit, T.S., Wray, R.E., & Folsom-Kovarik, J.T. (2015). Creating a Scenario Design Workflow for Dynamically Tailored Training in Socio-Cultural Perception. *Procedia Manufacturing, 3*, 1486-1493.

Keller, J.M. (1987). Development and use of the ARCS model of instructional design. *Journal of instructional development, 10*(3), 2.

Kocmánek, T. (2015). *HyperNEAT and Novelty Search for Image Recognition.* Master's thesis, Czech Technical University in Prague.

Lehman, J., & Stanley, K.O. (2008, August 5-8). *Exploiting Open-Endedness to Solve Problems Through the Search for Novelty.* Paper presented at the 11th International Conference on the Synthesis and Simulation of Living Systems (ALIFE), Winchester, UK.

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

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

Luo, L., Yin, H., Cai, W., Zhong, J., & Lees, M. (2016). Design and evaluation of a data-driven scenario generation framework for game-based training. *IEEE Transactions on Computational Intelligence and AI in Games*.

Reigeluth, C., & Stein, R. (1983). *Elaboration theory*.

Shute, V.J. (2008). Focus on formative feedback. *Review of educational research, 78*(1), 153-189.

Sottilare, R.A., Brawner, K.W., Goldberg, B.S., & Holden, H.K. (2012). The generalized intelligent framework for tutoring (GIFT). Orlando, FL: US Army Research Laboratory Human Research & Engineering Directorate.

Stanley, K.O., Bryant, B.D., & Miikkulainen, R. (2005). Real-time neuroevolution in the NERO video game. *IEEE transactions on evolutionary computation, 9*(6), 653-668.

Wray, R.E., & Woods, A. (2013). *A Cognitive Systems Approach to Tailoring Learner Practice.* Paper presented at the 2nd Advances in Cognitive Systems Conference, Baltimore, MD.

Zook, A., Lee-Urban, S., Riedl, M.O., Holden, H.K., Sottilare, R.A., & Brawner, K.W. (2012, May 29-June 1). *Automated scenario generation: toward tailored and optimized military training in virtual environments.* Paper presented at the International conference on the foundations of digital games, Raleigh, NC.

ABOUT THE AUTHORS

***J.T. Folsom-Kovarik, PhD** is the lead scientist at Soar Technology, Inc. for adaptation and assessment within intelligent training. He earned a PhD in computer science from the University of Central Florida in 2012. His research combines modern data science and machine learning approaches with SoarTech's long experience in modeling expert knowledge and human experience. When expert knowledge guides machine learning and data analytics algorithms, they become more applicable and useful in real-world training settings. The combination of approaches can remain feasible when available data is small, concepts evolve over time, or nontechnical users need to control the training.*

***Keith Brawner, PhD** is a researcher for the U. S. Army Research Laboratory's Human Research & Engineering Directorate (ARL-HRED), and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). He has 12 years of experience within U.S. Army and Navy acquisition, development, and research agencies. He holds a Masters and PhD degree in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida. His current research is in ITS architectures and cognitive architectures. He manages research in adaptive training, semi/fully automated user tools for adaptive training content, and architectural programs towards next-generation training.*