ABSTRACT

Scenario-based training (SBT) is an instructional approach ideally suited to take advantage of the characteristics of simulation. However, it is often difficult for instructors to create scenarios for computer-based simulation; this problem has prompted researchers to begin exploring automated scenario-generation techniques. Yet, for software to automatically generate an efficacious training scenario, it must “understand” the components, and the relationships among components, that comprise effective scenarios. This chapter discusses an approach to operationalizing scenario pieces so that automatic generation software can assemble complete training scenarios from these component parts.

Keywords: Scenario-Based Training, Scenario Generation, Training Objectives, Domain Ontology

INTRODUCTION

Developing scenarios for simulation-based training can be a time-consuming and expensive process. Unfortunately, such inefficiency often results in only a small
number of scenarios being developed and constantly reused. Not only does this reduce variety in training, but the loss of variety and limited scope of scenarios also diminishes training effectiveness (Martin et al., 2009). To help address these issues, the authors are investigating processes and tools for improved manual scenario generation, as well as computational approaches for automated scenario generation.

Yet, before any method for scenario generation can be implemented, the scientific community must truly understand what a training scenario is. This understanding includes not only parameters of the scenario, but also a conceptual model of scenarios, an operational definition of scenario complexity, and a framework for linking scenarios to learning objects. Objective, operationalized definitions of these facets will enable software to generate appropriate scenarios for specific trainees, based upon input such as their past performances and their individual characteristics and goals.

This chapter describes the foundational work of developing this conceptual definition of training scenarios. The conceptual approach is based upon the notion of selected “training objectives,” coupled with the use of “baseline scenarios” and modifications that offer increased complexity. The training objectives are enumerated based on the trainee audience and provided for user selection. Baselines represent simplified scenarios that depict ideal conditions (no weather, perfect lighting, no surprises, etc.). Additional modifications allow the baselines to be altered and the overall scenario complexity enhanced. Given a trainee profile, the system can assemble combinations of baseline and modifications that support specific training objectives, reach appropriate levels of complexity, and provide adaptive training opportunity for the trainee(s) to further their understanding and performance.

In addition to basic scenario generation based on trainee profiles, the science of training suggests that training scenarios should support varied pedagogical approaches. For example, a given trainee may be presented with a “compare and contrast” scenario where a particular event is specified in two different ways. Or a “disequilibrium scenario” may show a possible worst-case scenario to enhance the understanding of why such training is necessary. The authors will discuss how the scenario generation framework supports such capabilities, and example instructional strategy use cases will be presented.

Finally, the authors will briefly touch on some of the computational approaches that could be used to support automated scenario generation based on the presented conceptual model. A recently re-discovered approach and a newly-developed approach show great promise in their possible use in automated scenario generation. Both of these computational approaches fit particularly well with the proposed conceptual model, and their benefits and limitations will be outlined.

**AUTOMATIC SCENARIO GENERATION**

For the purposes of this research, scenario generation describes the design and development of training episodes for simulation-based instruction. Traditionally, this process is carried out by one or more subject-matter experts, who manually plan out the scenario and then carefully program their plan into a simulation system.
Automatic scenario generation occurs when a computer executes (or assists with the execution of) the scenario design and development process (for a more complete review, see Martin et al., 2009). Yet, what is a scenario? The answer to this question includes not only parameters of a scenario, but also a conceptual model of scenarios, an operational definition of scenario complexity, and a framework for linking scenarios to learning objects. Objective, operationalized definitions of these facets will enable software to generate appropriate scenarios for specific trainees, based upon input such as their past performance and individual characteristics.

**TRAINING SCENARIOS**

First, a common understanding of the term “scenario” is required. When discussing scenarios, a distinction must first be made between the scenario, itself, and the simulation in which it is embedded. Roughly, simulation supports simulation-based training, which is the use of a virtual environment to support practice. In contrast, scenarios support scenario-based training, which is often used in conjunction with a virtual environment. Thus, scenarios can be thought of as the purposeful instantiation of simulator events to create desired psychological states (Martin et al., 2010).

Next, the difference between a “scenario” and a “situation” must be clarified. Situations refer to instant snapshots, which occur at any given time within an exercise; whereas scenarios can be thought of a series of situations over time (Tomizawa & Gonzalez, 2007). Consequently, a training scenario is a series of simulator events that create specific situations, which facilitate situated learning. Finally, in addition to simply describing the environmental context, training scenarios should include pedagogical accompaniments, such as training objectives and performance measures, in order to facilitate optimal transfer-of-training.

**INPUT-PROCESS-OUTPUT**

Automated scenario generation can be conceptualized via an Input–Process–Output model (e.g., Hofer & Smith, 1998). Inputs may include specific training objectives and information about the trainees, i.e., information used to “seed” the generation process. Once inputs are received, the software processes them, and then assembles a scenario, constrained by certain predefined heuristics. The software then outputs a composite scenario definition file.

Specifically, the inputs include a preselected training objective, an optional recommended pedagogical approach, and information about the trainees, including the number of trainees, the functional roles they will play in the simulation, and their levels of expertise. Once inputs are provided, the generation system constructs a unique, valid scenario that emphasizes the given training objective and is tailored to the specific trainees’ instructional needs. The output composite scenario definition is automatically assembled from pre-existing scenario baselines, “vignettes” that represent an element of a scenario, and pedagogical templates. Specific instances of each are selected based on their goodness-of-fit relative to the
training inputs. The scenario is then output and used within the simulation systems for appropriate initialization of the exercise. Conceptually, XML is a good technology to represent the scenario components as it maximizes flexibility and is extensible to different simulation platforms.

SCENARIO BUILDING BLOCKS

In this section, the specific building blocks of a scenario (briefly discussed above) are articulated in more detail. All together these blocks are referred to as facets of the scenario; however, each has a specific role to play in the formulation of a scenario.

TRAINING OBJECTIVES

The military formally defines training objectives in their Training & Readiness (T&R) manuals. Each objective in the T&R manual is accompanied by a list of “conditions,” which describe the context under which the action can be performed. In the automated scenario generation system, these conditions become requirements for elements that must be present in the scenario. For example, to train an artillery gunner to fire upon an enemy convoy, the virtual environment must at least include available supporting arms, munitions, and an enemy convoy to target. Thus, the selection of a particular training objective causes a set of conditions to become “active” (i.e. valid for use in this scenario).

Note that training objectives can be created for any particular domain. While our initial focus is in military exercises, this approach could be used to develop scenarios for other domains such as cognitive rehabilitation or education. However, for any domain, it is important to consider what the training objectives for that domain would be and how it can be specified at a level appropriate for scenario generation.

For example, the training objectives in the military’s T&R manuals typically have broad definitions. For example, the Marine Corps Infantry T&R Manual describes training objective “0302-FSPT-1302: Employ Supporting Arms” as:

Given a radio, call signs, frequencies, available supporting arms, equipment, a scheme of maneuver and a commander's intent... achieve desired effect(s) on target that support(s) the ground scheme of maneuver.

Such a description is not sufficiently detailed for automatic scenario generation. Consequently, an approach must be devised to break down training objectives into their component knowledge, skills, and attitudes (KSA s) (see Fowlkes et al., 2010). Training objectives are decomposed into a formal “domain ontology,” which resembles a tree-diagram. For instance, the “Employ Supporting Arms” training objective may include KSA s related to spatial and temporal coordination, battlefield sense making, tactical positioning, and communication (just to name a few). KSA s
are associated with one another in a hierarchical fashion, with those KSAs that require more coordinated actions listed higher on the diagram.

**Baseline Scenarios**

Identification of a training objective triggers the construction of a baseline scenario. Baseline scenarios are simplified scenarios that include minimal, ideal conditions that support the selected training objective. “Minimal” implies that only those elements required to support the training objective are present. “Ideal” implies that all scenario variables are set to their least complex settings; for instance, weather conditions are set to optimal (e.g., daytime lighting, no precipitation, no wind).

Baseline scenarios can support training, in theory. However, these simple scenarios do not offer particularly beneficial training experiences. At best, they are only suitable for training the most novice trainees in procedural operations. They lack variability and complexity, and they are not well suited for the training of cognitive skills. In order to expand baseline scenarios into more functional training tools, additional scenario elements are needed.

**Augmentations**

The baseline scenario can be enhanced by adding augmentations. These elements add complexity to the scenario and can affect aspects of the scenario itself (both entity and weather). Examples include moving the scenario to night, adding an additional target, or making a target hidden from first view. Each of these adds complexity to the generated scenario; however, not all may be applicable to a given set of training objectives. In addition, some may need to be limited. It makes no sense to add the “night” augmentation to a scenario more than once. Therefore, specifications of augmentations must also provide some limitations on quantities allowed.

The adding of augmentations to a baseline begins to collect together an appropriate training scenario. Through adding elements to a scenario, appropriate complexity levels can be supported for the full range of trainees from novices to experts. Much like training objectives, augmentations add requirements to the scenario. Adding an additional target requires that the target be specified (type and position). However, it is important to note that augmentations still focus on the initial situation of the training exercise.

**Scenario Vignettes**

The baseline and augmentations set a basic initial situation, focused on the environment. In turn, scenario vignettes add learning-objective focused content to the baseline. Scenario vignettes are pre-packaged alterations and/or additions to the scenario; they may be both macroadaptive (i.e., predesigned to meet instructional needs of trainees) or microadaptive (i.e., adjusting in real-time to better facilitate the
Scenario vignettes are defined as sets of associated triggers and adaptations. Triggers are any kind of check or comparison that returns a Boolean (true or false) value. They may listen for specific events (e.g., a detonation occurred nearby) or be time-based (allowing time-specific events to occur if desired). When a trigger is determined to be true, its corresponding adaptation is activated. Note that a trigger could have more than one adaptation associated with it. In addition, triggers can be chained to provide “if-then” type logic or even Boolean “and” logic.

Adaptations are alterations made to the current situation within the exercise. They could include entity manipulations (create, kill, move, fire weapon) or environmental manipulations (reduce rain, raise sun). They may be used to adjust the focus of the training (e.g., providing remediation) or they may be used to repair an exercise. For example, if a critical entity is killed, then an adaptation can be used to recreate it and, thereby, facilitate the completion of the scenario (i.e., the training opportunity is not lost). Ultimately, adaptations cause changes to occur within the scenario itself. Together with triggers, they form the basis for adjusting scenarios during run-time (i.e., microadaptation).

While triggers and adaptations form the basis for a vignette, they also provide an additional capability to support dynamic scenario (training) adaptation. This form of scenario adaptation can be paired with machine intelligence or developer-prescribed rules to create new triggers and adaptations on the fly. In other words, in addition to a scenario creator (whether human or automated program) creating vignettes for anticipated events, an adaptation system can dynamically create new triggers/adaptations to satisfy common instructional needs during run-time. For example, if a trainee accidentally gets killed during an exercise, the system may have implicitly prepared a trigger to detect this and then fire an adaptation to respawn that trainee (so that the remainder of the training opportunity is not lost). Similarly, if trainees' performance falls outside of a predetermined range, then the system may trigger an adaptation that escalates the training (e.g., introducing the next training objective) or offers remediation.

SATISFYING REQUIREMENTS

Before the scenario can be considered complete, one additional step is required. The training objectives, baselines, augmentations and vignettes may have specified required elements for the scenario. For example, a vignette may require an entity to exist (such as a target). However, these scenario components generally do not specify the type or position of said entity. These details are left to the end.

Requirements can be satisfied either manually, by a user, or via an automated approach. We enumerate the types of requirements that may be necessary, which allows the system to prompt for the specification of that requirement type. For example, knowing that a target is required, the system can prompt for type and position of that entity. Once requirement is satisfied (specified), then the scenario is complete.
INSTRUCTIONAL STRATEGIES

In the previous subsections, scenarios were formalized as a baseline, a set of augmentations, and a set of vignettes (each made up of a set of triggers and a set of adaptations). To transform these elements into training scenarios, pedagogical data should be incorporated.

The science of training suggests that training scenarios should support varied instructional approaches. For example, a given trainee may be presented with a “compare and contrast” scenario where a given event is shown in two slightly different ways, which emphasizes the subtle differences between the cases. Or a “disequilibrium scenario” may show a possible worst-case scenario to enhance the understanding of why such training is necessary. Such instructional templates help guide the construction, and potential microadaptation, of a scenario. This is the final (high-level) building block for our conceptualization of dynamic scenario generation and augmentation.

BUILDING TRAINING SCENARIOS

With the formulation of scenarios operationalized, an application can then be built to allow a user to develop a scenario and export it in a way that allows various training applications to initialize (and then execute) the scenario. The authors are participating in a research program to develop such an application with an initial focus on U.S. Marines Fire Support Teams.

The user first selects the trainees' expertise levels (which in turn sets the scenario's complexity range) and training objectives to be trained. Coupled with the trainee profiles and the KSAs related to the training objectives, the baselines, augmentations and vignettes are then filtered to only provide those that support training that desired configuration. The user then selects a baseline, with zero or more augmentations and zero or more vignettes to create a scenario of sufficient complexity (the system ensures that the resulting scenario is within the desired complexity range and requires the user to alter the scenario if not).

Future work will investigate how to provide some automation to the scenario creation process. The user will still provide the initial inputs, but the system itself will then formulate a baseline, set of augmentations and set of vignettes to couple into a scenario fulfilling that request. Currently, procedural models such as shape grammars and L-systems are being considered.

CONCLUSION

In this chapter, an operationalization of scenarios was given. When creating scenarios, it is necessary to formulate a fixed definition for a scenario that can satisfy building a variety of scenarios given specific training objectives. This is more important for creating such scenarios in an automated fashion. In addition, the use of pedagogical templates can further enhance the training effectiveness. They
provide for scenarios that can use different lesson types to enhance the learning experience. This further enhances the variety of scenarios that can be provided as well.

ACKNOWLEDGEMENTS

This work is supported in part by the Office of Naval Research Grant N0001408C0186, the Next-generation Expeditionary Warfare Intelligent Training (NEW-IT) program, and in part by National Science Foundation Grant IIP0750551. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ONR or the US Government. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES


