Trend-Aware Scenario Authoring: Adapting Training Toward Patterns from Real Operations

Mark G. Core[1](✉), Benjamin D. Nye[1], and Brent D. Fegley[2]

[1] Institute for Creative Technologies, University of Southern California, 12015 Waterfront Drive, Playa Vista, CA 90094, USA
{core,nye}@ict.usc.edu

[2] Aptima, Inc., 12249 Science Drive, Suite 110, Orlando, FL 32826, USA
bfegley@aptima.com

Abstract. An important prerequisite to trend-aware authoring is that scenarios be authorable and inspectable by instructors but also machine-readable such that authoring tools can assist with integrating real-world patterns into training. In this research, we use a semi-structured approach to authoring flight training scenarios in which textual descriptions of related scenario elements (i.e., happening at roughly the same time) are grouped together and assigned training objectives and phases of flight. This same representation can be used to represent real-world emergencies allowing their integration into scenarios for more realistic training. Such a representation is sufficient to support a recommender that ranks possible insertion points for real-world emergencies using constraints (i.e., the phase of flight of the emergency must match the phase of flight of the insertion point) and a ranking score. Our ranking score is currently based on matching training objectives associated with the emergency with training objectives in the scenario (i.e., training the same skills but using a more realistic example). The recommender is integrated into the scenario editor such that instructors can see the ranked injection points and modify the scenario by selecting one of these points.

Keywords: Authoring · Scenario-based training · Recommender

1 Introduction

Scenario-based training offers a powerful tool for trainees to practice skills in simulated or narrative analogs of a real situation, which should increase the transfer of these skills to real-life tasks. However, developing training scenarios and ensuring that they train the intended skills remains challenging and labor intensive. Stacy and Freeman [7] suggest Training Objective Packages (TOPs) as a way to address the challenge of creating opportunities for training objectives during live and simulated exercises. A TOP encodes the conditions necessary for

R. A. Sottilare and J. Schwarz (Eds.): HCII 2024, LNCS 14727, pp. 15–24, 2024.
https://doi.org/10.1007/978-3-031-60609-0_2

the trainee to meet the training objective and how to measure and assess trainee performance.

Formal representations such as TOPs can enable better record-keeping for scenarios (e.g., knowing which competencies they train) and promote reusability. Different TOPs can be introduced into the same base scenario to train different skills. However, reusability does not ensure relevance: training scenarios are often static (infrequently updated) until replaced at significant cost. A scenario that fails to reflect changes in training needs or operational problems erodes the advantages of scenario-based training to transfer to real-life situations. In this paper, we will explore a potential solution to faster scenario updates, by integrating data from real-world trends, events, and emergencies into a scenario-authoring approach inspired by TOPs. Although this research remains exploratory, our design-based investigation suggests trend-based recommendations can be aligned to real-life training plans and in some cases can offer a quick, drop-in replacement for less relevant training activities.

2 Background

Training Objective Packages (TOPs) are one type of a broader category of representation designed to determine when a training objective can be trained and how performance should be measured. As noted, a TOP specifies: a) necessary conditions for the trainee to meet the training objective and b) how to assess performance on the training objective. Conditions are encoded as "behaviour envelopes" specifying boundaries on conditions, such as spatial or temporal coordinates [7].

A related approach [4] builds upon the Total Learning Architecture [1], a U.S. Department of Defense (DoD) standard for learning ecosystems, by introducing experience events (xEvents). Like TOPS, xEvents encode the conditions under which training objectives may be achieved. TOPS and xEvents can be seen as attempts to improve upon Master Scenario Event Lists (MSELs) in which authors use spreadsheet or word processor documents to outline a plausible sequence of events for a training exercise. It is difficult to trace the origins of MSELs as they appear without citation in training across a variety of organizations (e.g., military, police and other emergency responders) in which teams must work together to accomplish goals in environments which are often unpredictable (e.g., [6] describes best practices for authoring MSELs for US Homeland Security exercises). MSELs have the advantage of being human-readable and thus understandable by instructors, role-players, simulation controllers and trainees. There is generally a clear connection between events (e.g., clearing a room) and training objectives (e.g., practice room clearing) whereas machine-readable scripts for virtual and constructive entities are black boxes in which training objectives may not be represented or included in decision making. The goal of TOPs and xEvents is to retain the ability for instructors to author and inspect this data while using a machine-readable format to facilitate integration with simulators as well as authoring tools.

Authoring tools are particularly important because they can assist instructors in modifying scenarios and in particular introducing new elements based on lessons learned from the field. Scenarios based on real problems can boost trainee engagement and generally training offers the strongest advantage when it aligns closely to the problems and conditions for applying the skills in real situations (e.g., [5] discusses the importance of realistic radio simulations mirroring problems that arise on the battlefield). There needs to be a constant updating of scenarios based on current conditions and emergencies from the field because real-life needs change over time. For example, pilots might train to land an aircraft in the desert for years, but suddenly missions are needed in arctic conditions or where rain storms are frequent. These high-level changes are reflected in specific issues and signals: a different warning code or light might appear to signal that a landing is unsafe and must be aborted.

Unfortunately, there is often a substantial lag between trends changing in the field and instructors tuning practice to better reflect real-life. Experts must first recognize trends and lessons learned from the field, and bundle them into reports. Then course developers and instructors must review these reports and find "teachable moments" such as a real-life malfunction that can be used to practice emergency procedures. Due to the many steps in the process, such updates are infrequent and often limited to major changes (e.g., new equipment models). This could be improved through the use of authoring tools to update training based on real-life "teachable moments".

3 Domain: Flight Training Based on Incident Trends

As part of a broader research effort called TOPMAST (Training Operational Performance via Measure Automation and Scenario-generation Technology), we investigated approaches to speeding up the process for updating training scenarios based on observations from the field, with an emphasis on aviation training. Aircraft are particularly well-suited to a trends-based approach, because they have a well-defined taxonomy of issues (e.g., fault codes), they have multiple types of trends (e.g., aircraft versions, flight routes, subsystem updates), and require operators to quickly diagnose and react to emergencies to maintain safety.

In this work, we studied Navy flight instructors who took a MSEL-style approach to authoring using word processors to generate scenario outlines (e.g., events that should occur and rough time guidelines) and only providing scenario-level training objectives. To address this challenge, we developed a scenario editor for flight training that explicitly represents the structures of these flights starting with individual scenario elements. Figure 1 shows an example with details specific to the aircraft obscured or scrambled in the case of fault codes.

1. **Event:** a normal flight event (e.g., communications, achieving takeoff) or an emergency (a fault code or description of the problem) which is listed under Event Sequence. Faults are highlighted in red and indented.

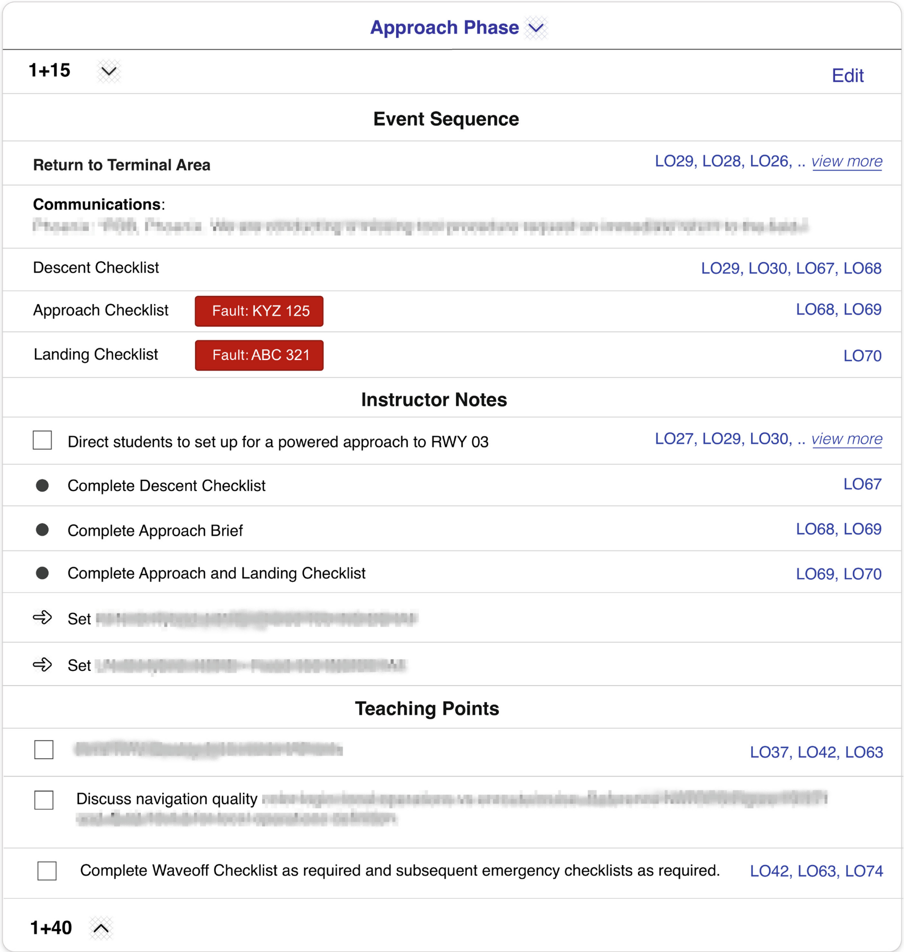

Fig. 1. Sample time point in TOPMAST scenario editor

2. **Expected Crew Action** (e.g., performing a checklist, responding to communications, manipulating controls) which is listed under Instructor Notes using round bullet points.
3. **Simulator Manipulation:** how the instructor triggers or resolves an emergency in the simulator which is listed under Instructor Notes using arrow bullet points.
4. **Instructor Action:** (e.g., giving verbal instructions to trainees) which is listed under Instructor Notes using square bullet points.
5. **Teaching Point:** item to discuss with trainees which is listed under Teaching Points.

Figure 1 also shows how scenario elements are grouped into a row and assigned a time point (e.g., after approximately 1 h and 15 min these scenario elements should occur). These rows are associated with a phase of flight which in this case is the landing approach. The scenarios under study typically assumed the presence of two trainees, with one having principal responsibility for control and safety of flight. Mid-way through the block of instruction, trainees would switch responsibilities, meaning that each scenario had two versions (one for student A, one for B). The student B version allows practice of the same training objectives but includes variations so that it is not a complete repetition. The scenario editor also includes a form in which authors specify initial conditions such as weather and aircraft state.

Following the guidance of our subject matter experts, we used the term "learning objective" in the scenario editor to reflect instructor expectations. In Fig. 1, we see scenario elements annotated with learning objectives (presented as hyperlinks) by the subject matter experts. Each learning objective is assigned a code prefixed by "LO" and authors use a drop-down menu with the full learning-objective names when annotating. In this paper, we will continue to use the term "training objective" to reflect their role as training goals (i.e., "the trainee should be able to do X").

In some cases, the mapping of scenario elements to training objectives is straight-forward and one-to-one (e.g., the crew action of completing the descent checklist is linked to the training objective of completing the descent checklist). In other cases such as a malfunction event, the scenario element can be linked to a set of training objectives (e.g., general troubleshooting, managing navigation issues, executing wave-off procedures). In current practice, the scenarios are well-defined, have clear training objectives, are limited in number, and static (i.e., training objectives for a course are rarely updated based on real-life events or a real-time data feed).

The overall TOPMAST system is designed to be a force multiplier. It is both a scenario library management system and a scenario generator. We envision that TOPMAST will support a set of official (approved) scenarios as well as allowing the creation of variants to keep training relevant and fresh, and mitigate the "gouge" whereby trainees effectively skip the decision-making process having memorized how to respond to scenario events. Version tracking will ensure that variants do not overwrite official scenarios and that the provenance of each scenario is known and preserved.

4 Recommender: Adding Real-World Events to Scenarios

To update training, data sources must be available to track real-world events. Data from the field typically is either an equipment log file or a written hazard/accident report. Equipment logs have the potential to be directly transformed into structured representations (e.g., database tables). However, although aircraft manufacturers may collect and archive such logs they are currently not readily available to training developers. Hazard/accident reports are distributed

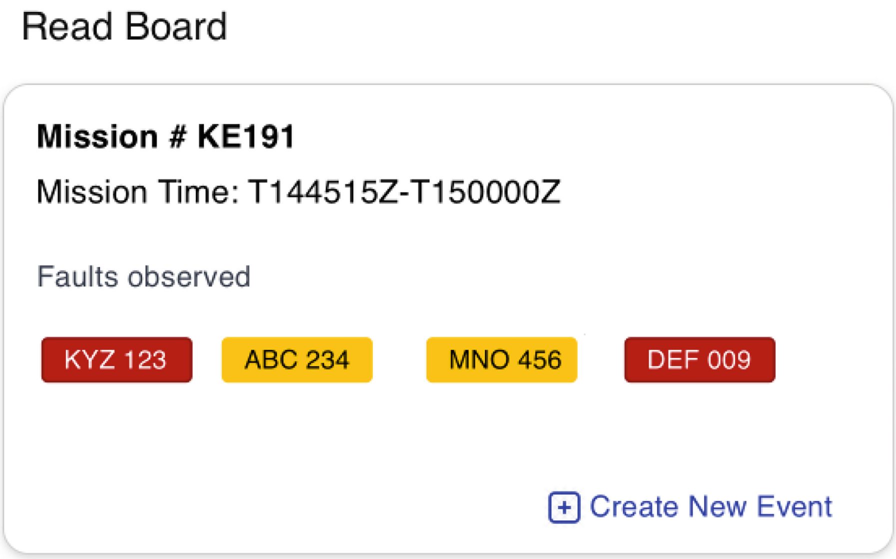

Fig. 2. Sample Read Board card in the TOPMAST scenario editor

widely and unlike equipment logs indicate causality and lessons to be learned. Each emergency needs to be described using the same format as the scenario (e.g., events, expected crew actions, teaching points, training objectives and phases of flight) which is currently done by subject matter experts. However, advances in natural language processing such as large language models (LLMs) should enable automation of the information extraction process for this domain, due to its relatively well-defined ontology of fault codes, events, and actions.

Once a new real-world emergency is authored in this way, it appears in the scenario editor's read board (Fig. 2). Once an author clicks "Create New Event" to add this emergency to the scenario library, the TOPMAST recommender system identifies and ranks "injection" points for this emergency into a library of training scenarios. Currently we analyze each row of the scenario (i.e., scenario elements occurring at roughly the same time) as a possible injection point, assuming the phase of flight of the emergency matches that of the scenario row. Such injections into a pre-existing scenario ensure the same skills are addressed but in more realistic conditions.

We then generate an injection score, currently the number of overlapping training objectives between the scenario row and the emergency. Such an approach is sufficient for matching real-world emergencies to scenario events that exercise the same training objectives. In future work, we intend to explore including expected crew actions, teaching points and related events in the injection score calculation. For an emergency occurring across two phases of flight, possible insertion points are constrained to these boundary points in the scenario

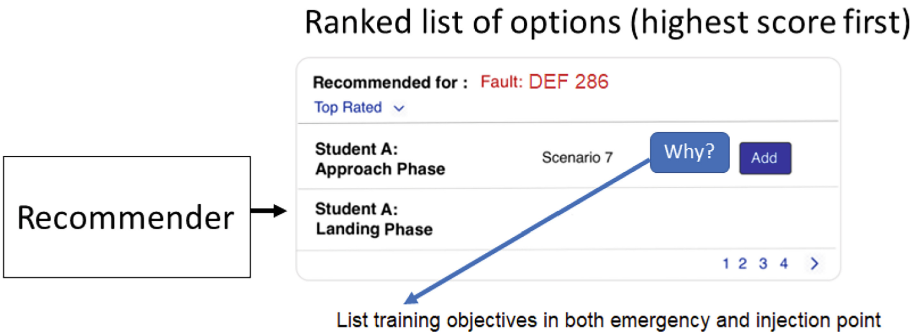

Fig. 3. Annotated mockup of recommender interface

library; injection scores for these boundary points can be calculated separately for each phase of flight and then summed.

Currently our recommender considers rows in isolation as possible locations to inject an emergency (i.e., only supports cause and effect in the same row). Once the author picks an injection point, the recommender appends the emergency elements to the row, which the author may need to modify subsequently. For example, in some cases the real-world emergency can serve as an explanation for an instructor-triggered fault and should immediately precede it (e.g., a fault in the navigation system could explain why the flight computer triggered a wave-off). In other cases, the real-world emergency should serve as a replacement for a fake event (e.g., a navigation sensor error triggers a wave-off instead of a vehicle blocking the runway). We will address this issue in future work by allowing authors to identify causal links (e.g., all crew and instructor actions related specifically to the vehicle on the runway) and enforcing the use of consistent vocabulary for checklists and procedures. In some cases background knowledge will be needed (e.g., fault A often causes fault B) to infer when a real-world event might be linked to a pre-existing scenario element.

The recommender is integrated into the scenario editor such that instructors can see the ranked injection points and modify the scenario by selecting one of these points. The interface is still a work-in-progress. Figure 3 is a mockup of how the interface would present an emergency impacting two phases of flight (i.e., approach and landing). The recommender maintains the set of common training objectives between the emergency and each injection point such that an author can inspect them before clicking the "add" button to inject the emergency.

5 Discussion: Scaling Up Trend-Aware Authoring

Training objective packages (TOPs) [7] are a general approach to authorable, machine-readable scenario representations. TOPs represent both the necessary scenario conditions for a trainee to achieve a training objective as well as the criteria for achieving that training objective. Our flight instructors took an approach similar to a Master Scenario Event List and used our authoring tool to specify a sequence of events annotated with training objectives. The conditions for a particular event and associated training objective are assumed to be the accomplishment of all the previous events (i.e., there is no representation of causal links between events).

In the context of Navy air training, a potential resource for this missing information is the Naval Air Training and Operating Procedures Standardization (NATOPS) manual for the target aircraft. Electronic versions of some NATOPS manuals have been available since 2002 [3] but focus on supporting pilots (e.g., quick access to emergency procedures) rather than representing knowledge for machine use. The rows of a scenario group related elements (e.g., a system fault may be in the same row as the corresponding wave-off) but in some cases, a minor malfunction may not cause an emergency until a phase of flight such as "approach" in which precise control and navigation are critical.

The introduction of flight recorder data would also present a number of opportunities. Stacy et al. [8] discuss a tool to analyze flight recorder data from an emergency and author a corresponding scenario event list. The availability of such data would also allow analysis of trends to suggest real-world malfunctions that are more frequent and identify possible training gaps. A trends-based approach would allow a human expert or a computational model to analyze data from many flights to measure malfunction frequencies and how they co-occur and vary based on conditions. This is especially critical with a newer aircraft as it could lead to changes in procedures and training such that early corrective actions can prevent more serious consequences.

Large language models (LLMs) such as GPT-4 have been used for natural language processing and common sense reasoning tasks [2] and could potentially aid the human author by processing text and making connections (e.g., event A causes event B). Newer LLMs could also help to extract real-life trends from noisier data sources (e.g., flight logs, mission reports). It may even be possible to suggest modified scenarios by having LLMs modify a standard flight to meet new training objectives.

Another opportunity would be the ability to integrate authoring tools such as TOPMAST with the simulator. Currently scenarios include guidance for instructors on how to manipulate the simulator to introduce faults to test trainees and then resolve those faults when the trainees have performed the appropriate procedures. Ideally, TOPMAST could serve as both an authoring tool and simulation controller such that it could handle such details and allow the instructor to focus on observing and guiding the trainee. More complex logic could be introduced to make the scenarios adapt to individual trainees (e.g., giving more difficult, realistic challenges to high performing trainees).

6 Conclusion

This paper describes an effort to introduce machine-readable, instructor-authorable scenarios to flight training currently using scenario outlines authored with word processors. A semi-structured approach was taken in which related scenario elements (i.e., happening at roughly the same time) were given textual descriptions, grouped together and assigned training objectives and phases of flight. This same representation can be used to encode real-world emergencies allowing their integration into scenarios for more realistic training. Such a representation is sufficient to support a recommender that ranks possible insertion points for real-world emergencies using constraints (i.e., the phase of flight of the emergency must match the phase of flight of the insertion point) and a ranking score. Our ranking score is currently based on matching training objectives associated with the emergency with training objectives in the scenario (i.e., training the same skills but using a more realistic example).

The recommender is integrated into the scenario editor such that instructors can see the ranked injection points and modify the scenario by selecting one of these points. Our subject matter experts indicated that this type of tool would be beneficial over the current approach of using a word processor to add real-world emergencies to the scenario outlines. Although aircraft equipment logs are not available at this time, it is an important issue to address in the future as instructors must now review hazard/accident reports and select representative emergencies rather than the recommendation system directly measuring trends and associated conditions (e.g., weather, equipment readings). Another important future consideration is integration with the simulators used in training which must be configured to trigger emergencies matching their real-world counterparts.

Acknowledgments. Research reported in this paper was supported by the U.S. Naval Warfare Center under award number N68335-19-C-0583, as a collaboration between Aptima, Inc. and the USC ICT University Affiliated Research Center (U.S. Army W911NF-14D0005). The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

1. ADL Initiative: Total learning architecture functional requirements document (2021). https://adlnet.gov/assets/uploads/2021%20TLA%20Functional%20Requirements%20Document%20w%20SF298.pdf
2. Bubeck, S., et al.: Sparks of artificial general intelligence: early experiments with GPT-4 (2023). https://arxiv.org/abs/2303.12712
3. Deaton, J., Glenn, F., Burke, C.S., Good, M., Dorneich, M.: Evaluation of an interactive electronic NATOPS (IE-NATOPS) and associated graphic interaction concepts. Final report, Office of Naval Research (2002)
4. Hernandez, M., Blake-Plock, S., Owens, K., Goldberg, B., Robson, R., Ray, T.W.F.: Enhancing the total learning architecture for experiential learning. In: Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) (2022)
5. Khooshabeh, P., Choromanski, I., Neubauer, C., Krum, D.M., Spicer, R., Campbell, J.: Mixed reality training for tank platoon leader communication skills. In: IEEE Virtual Reality (VR) (2017)

6. Renger, R., Wakelee, J., Bradshaw, J., Hites, L.: Steps in writing an effective master scenario events list. J. Emerg. Manage. **7**(6), 51–60 (2009)
7. Stacy, W., Freeman, J.: Training objective packages: enhancing the effectiveness of experiential training. Theor. Issues Ergon. Sci. **17**(2), 149–168 (2016)
8. Stacy, W., Picciano, P., Sullivan, K., Sidman, J.: From flight logs to scenarios: flying simulated mishaps. In: Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) (2010)