Training objective packages: enhancing the effectiveness of experiential training

Webb Stacy & Jared Freeman

To cite this article: Webb Stacy & Jared Freeman (2016) Training objective packages: enhancing the effectiveness of experiential training, Theoretical Issues in Ergonomics Science, 17:2, 149-168, DOI: 10.1080/1463922X.2015.1111459

To link to this article: http://dx.doi.org/10.1080/1463922X.2015.1111459

Published online: 02 Dec 2015.

Submit your article to this journal

Article views: 50

View related articles

View Crossmark data
Training objective packages: enhancing the effectiveness of experiential training

Webb Stacy and Jared Freeman
Aptima, Inc., 12 Gill Street, Suite 1400, Woburn, MA 01801, USA

ABSTRACT
Training objectives define the purpose of instructional events; attaining them is the measure of successful training. Yet, it is challenging to apply training objectives in large, complex, multiparty military exercises. In such events it can be difficult for trainers to determine which students were able to address their objectives in a given scenario-based exercise, or, in some cases, whether they were able to address any training objectives at all. A scalable, formal mechanism is required to document and manage training objectives, their relationships to scenario conditions, and the performance measures by which attainment of objectives is evaluated. In this article, we describe Training Objective Packages and two subordinate formalisms: behaviour envelopes, which specify the bounds on student behaviour given conditions, and a formal expression of performance measurements that includes an approach called measurement envelopes. Each of these has value in the three phases of training: planning, execution, and assessment. We define these formalisms and describe several applications and opportunities for research.

Relevance to human factors/ergonomics theory
The design of technology for systematic, scalable training is a long-standing human factors challenge in instruction. This challenge must be met for all training, but is particularly important for training the operators of new technologies and techniques that are often designed, in part, by human factors engineers. The present article describes formal representations of training objectives, training conditions, and measures that should enable human factors engineers and instructional designers to better plan, manage, and assess training.

In order for training exercises such as [Fleet Synthetic Training] to be effective and worthwhile, specific training objectives associated with relevant performance measures linked to mission essential tasks must be identified and incorporated into the training events. (Vincenzi et al. 2007)
Introduction

Military training exercises like those depicted in Figure 1 are designed to develop and test essential military skills on land, sea, and air. Such exercises may run for days, and engage tens or hundreds of participants. Many of these are trainees, but a surprisingly large number are responsible for managing the event and controlling training simulators. Increasingly, these exercises involve a combination of real people operating real systems (a ‘Live’ component), real people operating simulated systems (a ‘Virtual’ component), and simulated people operating simulated systems (a ‘Constructive’ component). When brought together, these comprise a Live–Virtual–Constructive (LVC) exercise.

Clearly, large military training exercises require a significant commitment of resources. To net a return on that investment, training scenarios for these events should systematically address well-specified training objectives. Often, they do not. This violates best practices learned from the research literature. Analyses of athletes, musicians, physicians, and other professionals find that expertise is primarily a function of practice and feedback that are deliberately designed to strengthen weak skills (Ericsson and Lehmann 1996). Experimental research establishes that instructional processes (vanLehn and Chi 2012) and materials (Schwartz, Martin, and Nasir 2005) that are crafted to address specific training objectives do reliably strengthen the performance on the target task (near transfer) and on new tasks (far transfer). For this reason, training objectives have long been a requirement of instructional design (Mager 1962; Gagne 1985) and of institutional training practices, as the opening quote concerning Navy training illustrates.

When scenario design, scenario execution, and assessment fail to address training objectives in large military exercises, it is for pragmatic, rather than philosophical reasons. First, due to the complexity, dynamic nature, and sheer scale of these exercises, training designers find it extraordinarily difficult to plan multi-participant scenarios to

Figure 1. Large military exercises, such as those depicted here, involve significant resources, both for planning and for execution. They are often so large and complex that it is difficult to provide specific focus on individual or team training objectives.
systematically address individual and team training objectives (TOs). Second, student actions during training often drive scenarios off course, far from the planned training objectives, and it is difficult for trainers to recreate the required training conditions within the flow of a running scenario. Similarly, it is difficult for trainers to detect and exploit emergent but unplanned opportunities to address training objectives. In short, it can be difficult to plan, manage, and exploit the training conditions required to fulfil training objectives. Third, it is challenging for trainers to assess training, to determine which students successfully addressed which TOs in a given scenario-based exercise, or whether they were able to achieve any TOs at all. These are failures of the three fundamental training functions shown in Figure 2.

Trainers have traditionally adopted two practices to handle these problems: systematic documentation of training plans and simulation scripts.

Plans for multi-participant training scenarios are typically documented in a Master Scenario Event List (MSEL), which are authored using spreadsheets or word processing applications. MSELs describe the conditions — the mission, environment, entities, and activities — that students will encounter. They sometimes identify the TOs associated with these conditions. These descriptions communicate intended scenario conditions among the people who plan, monitor, and control training exercises, but since they are not machine-readable, MSELs cannot drive the computer-based infrastructure with which training is developed and delivered.

Machine-readable scripts are an alternative or complement to MSELs when large exercises involve virtual and constructive entities in addition to live participants, as they increasingly do. Software scripts can drive simulators during the exercise, but they present their own challenges. Scripts necessarily represent the actions of human and computer-generated entities in the scenario. However, these scripts do not anticipate all the possible

---

**Figure 2.** Pedagogical strategy for scenario-based training. Training plans define the conditions under which students can demonstrate competency on training objectives, and the measures by which competency is assessed. Training delivers those conditions. Assessment evaluates student competency on training objectives. This is an ideal set of relationships for training, but, despite best intentions, is seldom achieved in large military exercises because of sheer size and complexity. This is the challenge addressed by Training Objective Packages.
scenario plays. It usually falls to human operators to actively manipulate the training conditions within the exercise to enable trainees to address the training objectives, or at least practise their roles opportunistically in a coherent environment. Recent innovations have enabled more flexible, event-based behaviour specifications for computer-generated entities, and these are robust to many varieties of student behaviour. Prime examples of this technology are the Behavioural Transition Networks (BTNs) within the Navy’s Next Generation Threat System (NGTS), and the Training Executive Agent (Wray et al. 2015) that supervise BTNs. However, in contrast to the human-readable MSELs, it is fair to say that script-based and event-based specifications are difficult for humans to understand, and they are not necessarily designed to represent TOs or to make them explicit.

Thus, the central challenge is to create experiential training that is reliably and measurably focused on training objectives. This will require a formal system that helps ensure TOs are addressed, that exploits emerging TO-focused training opportunities, that assesses how well the students have addressed their TOs, and that scales to complex scenarios and large student groups. The foundation of this formal system must be a mechanism for accurately and precisely expressing TOs, one that is understandable to humans and that is also understandable to automated systems.

To meet the challenge, in this article we describe an approach called Training Objective Packages (TOPs), which are designed to efficiently and expressively represent essential training constructs and to drive training technologies. A TOP consists of a description of the TO itself and formal descriptions of two related components: (1) the scenario conditions that define a training opportunity for the TO, and (2) human performance measurements. To express scenario conditions precisely but flexibly, we define behaviour envelopes (BEs) and a corresponding language called Activity Description Language (ADL). To specify measurements and assessments of trainee performance on TOs, we present the Human Performance Markup Language (HPML).

These two languages are readable and writable both by humans and by machines. They provide a solid basis for automated tools that leverage TOPs in planning, execution, and assessment phases of experiential training (see Figure 2). Further, TOPs scale from small training situations to large ones. Even with a small number of participants, instructors can be overwhelmed trying to monitor training conditions and trainee performance; the approach we propose will allow them to focus on the most relevant aspects of training. However, it is larger training events that will show the highest return on the investment. Gains in training effectiveness and efficiency will, we believe, well outweigh the cost of deliberate instructional design, developing the computational representations recommended here, and applying the software to compute over them.

Training Objective Packages

TOPs are designed to capture key aspects of training in a scenario. Each TOP is a trio of

1. a TO, named and optionally described and located within a hierarchy of TOs;
2. the scenario conditions required to enable the student to make progress against the TO; and
3. measures of student performance relevant to the TO given the conditions and assessments of the student proficiency with respect to the TO.
Because the TOPs are extensible, they can eventually include other components, as we discuss in the Current and Future Research section.

TOPs are useful in experiential training whether or not the TOP and its components are represented formally or there is an automated system involved. For TOPs, automation is an enhancement, not a requirement. In subsequent discussions of TOPs in this article, our focus will be on the formal description of TOPs and the use of TOPs in an automated system connected to the training environment and a system that monitors scenario conditions and performs automated performance measurement. Combined, formal representation and automation provide major benefits.

Examples of TOPs

A few examples of TOPs in action may illustrate their value. Because these examples are necessarily verbal descriptions, they may seem to resemble MSEL events. However, we submit that (1) they can be represented formally, as we discuss shortly, and (2) they systematically include both scenario conditions and student performance measures, which are not true of all MSELs. We begin with two TOPs concerning planned training objectives:

(1) For military helicopter pilots, buddy lasing requires that operators in one helicopter point a laser accurately at a target, to generate data that are used by a second helicopter that fires on the target. For a TO related to buddy lasing, a TOP might represent that TO and the TO conditions, which require three entities: a helicopter that indicates the target with a laser, a helicopter that fires a missile at the target, and a target. These entities need to be arranged with certain geometric factors (angles, distances, etc.) with respect to each other, appropriate sensors need to be engaged, and there may be constraints on weather. The TO might also require that there be distractor entities in addition to these three engaged to make the training realistically complex. For TO measurement, the TOP might represent user accuracy at lasing and firing, and speed of use of the laser and the missile controls.

(2) Training in the use of sonobuoys, a technology for submarine detection, requires that technicians place these sensors well. For a TO related to sonobuoy management, the TO conditions involve signals on the student’s sensors indicating the possible presence and location of a submarine and deceptive behaviour on the part of the submarine. TO measurement might then involve the location and number of sonobuoys dropped in acceptable locations and numbers as well as the speed of identification of the submarine track. Assessments would categorise student performance according to the distance from ideal sensor locations and the speed with which they identify the track, relative to an expert.

It might happen that these TOs, though planned, are not addressed during scenario execution. The lasing helicopter in (1) may be destroyed early in the training scenario and cannot be resurrected in time to encounter its partner and target. The submarine operator in (2) may pilot to an unexpected position, far out of sensor range, and it cannot be replaced by a synthetic entity within the range of the sensor. Automated or human trainers using TOPs can identify that TOs are not addressed because conditions do not allow.

TOPs can also help identify teachable moments, that is, conditions in which participants can address TOs that are relevant to their role or mission, but that were not the
designed focus of the training event. These unplanned, emergent TOPs will not appear in
an MSEL, because the MSEL defines the planned events only.

(3) When a military pilot in an exercise inadvertently flies into an enemy Weapons
Engagement Zone (WEZ), the instructor may choose to take advantage of the situa-
tion to teach the pilot how to detect the event and take actions that pre-empt or
defend against enemy attack. This is an emergent TO. Its activation might cause a
synthetic enemy unit to fire a simulated weapon at the student, and measures of
student response to be taken, even though it was never planned that the student
venture into the WEZ. The TO would involve escaping the WEZ, the TO conditions
would be that the student is in the WEZ and a missile is fired, and the TO measures
might gauge how quickly and successfully the student exited the WEZ. TOPs can
represent these things.

(4) The operators of Unmanned Aerial Systems (UAS) must learn to coordinate with
each other to ensure that moving targets stay within the sensor range of one UAS,
or are handed off to other UAS as they move out of range. An emergent teamwork
TO might involve appropriate backup behaviour. Suppose, in a complex exercise,
that a UAS operator loses contact with an important surveillance target. This loss
of contact was not planned. If this TOP is monitored during the exercise, the
instructor will discover this emergent opportunity, and might assign the backup
behaviour TO to a teammate. The TO conditions would be that a team member has
lost contact with a surveillance target, and the TO measures would involve whether
the two operators exchange required information, re-establish contact, and do so
within some timed window of opportunity.

Such emergent opportunities occur frequently in large and complex exercises, and in
free-play training (in which TOs are not planned). Instructors may intuitively recognise
them, but it is fair to say that automated TOPs monitoring for emergent opportunities will
enable trainers to attend to other instructional tasks until these opportunities arise, alert
them reliably, and automatically apply preselected measures and assessments. Of course,
when an unplanned training opportunity occurs, instructors may choose either to take
advantage of the ‘teachable moment’ or to keep the student focused on current objectives.

Benefits of TOPs

TOPs support the three essential functions of training in Figure 2. During planning, TOPs
make training objectives, required conditions, and measures explicit among trainers. This
potentially increases the number of TOs covered by a planned scenario, even when the
number of TOs is large, and the TOs vary across many participants in the training event.
This is illustrated by planning examples (1) and (2).

During execution, TOPS simplify the monitoring of training objectives. They enable
the trainer and training system to detect whether training conditions satisfy planned
training objectives or not (examples (1) and (2)), and detect whether unplanned condi-
tions set the scene for addressing unplanned but relevant training objectives (examples
(3) and (4)). These benefits are illustrated in Figure 3.

During assessment, the records of monitored training objectives are available to train-
ers, to assess which TOs were addressed as planned, which were missed, and which
unplanned TOs were addressed opportunistically.
The net effect of TOPs is to make training plans more rigorous and expressive; make training and measurement opportunities more frequent, reliable, and responsive; and make assessments more informative. These are hallmarks of good training design, training delivery, and training evaluation.

TOPs achieve these benefits through several novel features:

- TOPs formally represent fundamental elements of training: TOs, conditions, and performance measures. The languages involved (ADL and HPML, described later) are readable and writable both by humans and by machines. Thus, training plans are richer and more actionable both by trainers and technologies. TOPs use the same formalisms for planned and unplanned training objectives. Thus they can both be identified as present (or absent), monitored, measured, and assessed consistently. That information about planned, missed, and unplanned training opportunities is available for after-action review (AAR).

- TOPs predefine measures to be taken and assessments to be made when those scenario conditions occur. Thus, there is consistency in performance measurements and assessments, and they are always linked to one or more TOs.

- TOPs contain all the data required to parameterise special purpose technologies that we define later. These are BEs, which manage training conditions relevant to training objectives, and the HPML, which manages performance measures relevant to training conditions.

Figure 3. Training Objective Packages provide more opportunities for training and performance measurement. (a) The actual trajectory of a student through the scenario does not always match the planned trajectory, and this can result in missed training and performance measurement opportunities. (b) Training Objective Packages specify only the required scenario conditions, leaving all others unspecified, and therefore increase the chances for the student to encounter training and performance measurement opportunities. They also provide for the identification of unplanned but potentially valuable ('teachable moment') training opportunities.
Managing training conditions using behaviour envelopes

BEs formally specify the scenario conditions required by a TOP. In contrast to scripts, which prescribe one stream of events and behaviours, BEs define boundaries on scenario conditions. Entities are free to take any trajectory ‘within’ the BE. For example, humans or computer-generated entities may want to work around obstacles to accomplish goals. Of course, entities sometime venture ‘outside’ the BE, in which case they are said to violate the BE. When used in TOPs, BEs codify ‘conditions for learning’ (Gagne 1985), that is, they express circumstances in which the student can and should exercise the knowledge and skills required to fulfil training objectives. They may also specify other factors that shape the behaviour of entities less directly, such as weather.

BEs can specify the conditions for students, human role-players, or computer-generated entities. For students, staying within the BE means that they have the intended scenario experience. For human role-players, staying within the BE constitutes their tasking for the exercise. For computer-generated entities, BEs effectively parameterise or constrain the mechanisms that create specific trajectories through the scenario. Even static scripts and event-driven approaches can benefit by becoming BE-aware: they can implement some other behaviour when the entity reaches a boundary. But generative approaches, those whose behaviour evolves with the evolving scenario context, as are typically implemented by intelligent agents (for example, Jones, Laird, and Nielsen 1998), benefit the most; they can, for example, seek alternate paths around an obstacle in pursuit of a goal, as long as they stay within the BE. Further, for all computer-generated entities, BEs can serve as a basis for validation of their behaviour, to the extent that the entity stays within the BE during scenario execution.

The simplest envelopes are spatio-temporal, specifying the range of places the instructor intends or for an entity to be at a range of times. Figure 4 depicts this concept graphically. The volumes depicted in the figure are defined by their BEs. Note that the BEs can provide more scenario leeway than scripts, which typically specify exactly where an entity should be at a given time, and exactly how it should behave (e.g., the specific route an entity should fly).
aircraft should fly at a specific time). However, BEs can constrain behaviour as tightly as a script, should that be what is called for by the TOP.

Bounds or ranges are expressed as sets of constraints (Stacy, Walwanis, and Colonna-Romano 2007). A constraint is simply a mathematical relationship between variables or between variables and constants — for example, asserting that an entity’s position is within a certain polygon on the map, its speed is less than Mach 1.0, its distance from target is greater than .5 miles, its altitude is greater than the nearest enemy aircraft, and so on. A constraint may be so narrow that it is effectively a specification of a single acceptable behaviour (e.g., configure radio to channel 7); thus envelopes provide the same functionality as scripts, when that is necessary, but offer considerably more freedom in general. A MSEL event that has traditionally been specified by putting a specific entity in a specific place at a specific time doing a specific thing can now become a specification of only those aspects of the event that matter, leaving all other aspects unspecified. ‘UAS Bravo arrives at the designated waypoint at 15:22:13 into the scenario and tracks ground vehicles Charlie and Echo’ becomes ‘Before UAS Sierra is on station, UAS Bravo tracks two or more ground vehicles simultaneously.’ By loosening the scenario conditions like this, there is considerably more opportunity to realise these conditions during scenario execution, and there is reduced worry that UAS Bravo cannot get where it needs to be on time. Further, the looser specification is concise; it may be all that is needed to provide the crew of UAS Bravo with the opportunity to work on training objectives relating to tracking ground vehicles. If more complexity is required, the parameters of that complexity can also be specified for the BE.

BEs can involve any variable, not just spatio-temporal ones. For example, they could involve fuel state, weather conditions, human workload, enemy intent, length and vocabulary of communications, cultural sensitivity, and a world of others. Envelopes may involve a single entity or may involve the interactions among multiple entities. In the latter case, the unit of envelope specification is the ‘vignette’ rather than the entity itself.

A language called ADL specifies envelopes formally. ADL is a language defined by an XML Schema, which means it is readable by machines and interpretable by humans.

The constraints involved in describing BEs in ADL map directly to the constraints used in mathematical constraint programming (e.g., Apt 2009; Dechter 2003; van Hentenryck and Michel 2009), a mature and successful subfield of Artificial Intelligence for solving combinatorial search problems like resource allocation and scheduling. Constraint programming specifies the problem in a declarative way: users can state the conditions that a solution must satisfy without having to specify the steps required to find the solution. This eases the process of eliciting them from instructors or other subject matter experts. We envision an instructional system that asks instructors what TOs are important for a given training session, and what scenario and training conditions are necessary for each TO. These could include the presence or absence of other entities and behaviour, events that should have happened beforehand, skills that should have been acquired beforehand, and a wide variety of other potential conditions. The system can then express these conditions in ADL, which can in turn be used for planning and monitoring the scenario.

The major elements of ADL are shown in Figure 5. The elements of constraint are one or more variables with a specified relationship. Typical relationships are simple
mathematical ones such as ‘≤’, ‘=’, and ‘≠’; however, the relationships can also be more complicated, such as the relationship ‘all different’ among multiple variables. Importantly, a variable has a domain which represents the possible values it can assume. Domains can consist of a finite set of integers, such as the number of threats a student should face during a vignette, a continuous interval, such as a range of distances from a target the instructor intends for the student, or a finite set, such as the defensive resources a student can bring to bear. Domains can be refined during the course of scenario execution as resources change and time elapses. Variables and domains are defined in the support element, and constraints are defined in the state element of BEs.

A constraint programming approach is ideal for planning and replanning events in a scenario, since it can optimise things like the number of training opportunities in the
scenario or the length of the scenario. In addition, the declarative nature of constraints makes them straightforward to monitor in real-time during scenario execution, and this can lead to an understanding of the conditions of training actually encountered by the student as well as the identification of unplanned training opportunities.

There have been other approaches to training that exploit constraints, but they have taken a somewhat different approach to defining constraints. Mitrovic and Ohlsson (1999) developed a constraint-based intelligent tutoring system to teach structured query language (SQL), a common database query language. More recent efforts (Wray and Woods 2013; Woods et al. 2015) have applied a similar approach to ill-defined tutoring domains. In both cases, their motivation was to develop a way to represent correct student behaviour without having to specify all the possible responses the student could give; students only needed to avoid violating the specified constraints on their answers. There is an underlying similarity between this usage and the present discussion of BEs in the recognition that constraint-based specifications can capture a large number of behaviours with a small number of constraints. However, their approach was focused on the specification of student responses instead of scenario conditions, and their definition of constraints focused more on constraints as rules and less on the mathematical view of constraints. We discuss the use of BEs in performance measurement further below.

One of the most challenging aspects of the formal specification of BEs is the specification of interactions among entities. This is not yet a part of ADL, but we are working to include it. The challenge arises because there are multiple kinds of coordination possible among entities. We have been able to express simple coordinated activities by leveraging concepts from computer science for coordinating activities among threads executing in parallel (Stacy, Colonna-Romano, and Roberts 2009). In addition, we have discovered that it is difficult to specify the interactions from ‘inside’ the envelopes for each entity involved in the interaction. The constraints are much more gracefully specified at the vignette level. Our intent is to maintain the constraint-based specifications so that they remain compatible with mathematical constraint programming, but to add an interaction element to the vignette element that captures the desired coordination among entities.

**Benefits of BEs**

BEs have value across the phases of training. During planning, designers can use BEs to define expectations of student behaviour given the conditions they encounter in training. During training execution, BEs can be used to automatically monitor the behaviours of students and synthetic entities. Trainers can then identify those students who need help. Systems can automatically guide synthetic entities towards behaviours that are instructionally useful (not just tactically realistic). During assessment, BEs can document whether planned training conditions were achieved, and this can help instructors to account for the success or failure of training to achieve its objectives or of students to achieve passing performance assessments.

BEs deliver this functionality largely because of the following reasons:

- BEs formally represent the bounds on (un)desirable behaviour given training conditions. This generally improves the clarity of training plans, and the quality of control of training events.
• The use of bounds makes BEs generalisable across training scenarios or across different plays of a single scenario. Thus, BEs present an efficiency proposition to instructional designers, who can build libraries of reusable BEs for a given domain.

Measuring and assessing performance

TOPs identify measures, and it is with these measures that a training system or an instructor determines whether a student satisfies training objectives and how (by what actions) they did so (Paris, Salas, and Cannon-Bowers 2000). Measures can serve a variety of functions other than evaluation, of course. Measurements are used as feedback, which is required to attain mastery (Ericsson and Lehmann 1996); as feedforward to alert students to practise and learning opportunities; to guide selection, adaptation, or generation of instructional content; to control the events, environments, and entities that constitute the conditions for learning; and to evaluate the effectiveness of training itself in ways that can improve curricula and instructional design (Freeman, Stacy, and Olivares 2009). Here, we describe formalisms for specifying measures, so that measurements can be made and used in these ways.

We make a distinction between measurements, which are direct computations on the raw data, and assessments, which are computations on the measurements and the data that provide interpretation for measurement. For example, a student’s answers on an exam are the raw data, the corresponding per cent correct on the exam is a measurement, and the conversion of the per cent correct into a letter grade is an assessment. A pilot’s glideslope position as he or she lands the plane is the raw data, computation of the deviation from ideal glideslope during landing is the measurement, and categorisation of the size of that error as acceptable or unacceptable is the assessment. A leader’s electronic communications with her distributed team are the raw data, the patterns inherent in those communications are the measurements, and the extent to which her communication patterns match the communication patterns of known high-performance leaders is the assessment. For TOPs, we will need an overall assessment to know whether the student has sufficiently mastered the TO, but other assessments as well as measures will help diagnose difficulties the student may be encountering with respect to the TO.

Human Performance Markup Language

There has been recent interest in standardising human performance measurement in training, and one result has been a push towards a standard way to specify the definition of and results from the computation of human performance measurements and assessments called HPML. HPML (Walker, Tolland, and Stacy 2015; Wiese et al. 2012; Stacy et al. 2005) is currently being discussed in a Study Group of the Simulation Interoperability Standards Organisation (SISO), which is the first step on the way to standardisation.

Figure 6 shows portions of HPML that are relevant to the present discussion. At the top level are Measures, Assessments, Results, and a variety of other related key elements. The figure also shows some detail for Measures and Assessments. These are actually templates that are used in the process of defining measures and assessments. The definitions themselves are shown in Measurement Definitions and Assessment Definitions, and the usage of
the definitions in a specific scenario is captured by the *Instances* element. Measurement and assessment outcomes are captured in the *Results* element.

HPML can also define TOPs. In HPML, TOs are described in the *Training Objective* element. TOPs descriptions are captured in the *Training Objective Packages* element, shown in Figure 7. The structure of that element is exactly the structure described in this paper — a TO together with the scenario conditions that represent a training opportunity for that TO and the measures and assessments related to that TO. Currently though, the description of conditions is limited to a text description. In the future, we hope to integrate ADL into HPML so that the conditions for TOPs can be described using BEs.

**Measurement using measurement envelopes**

One kind of measurement that is not yet explicitly represented in HPML is measurement envelopes (MEs). MEs are an additional method for measuring performance, and when added to HPML will be a kind of *System Measure*, which is one class of HPML measure as referenced in Figure 6. MEs are simply BEs used for measurement. MEs define in detail
the measures with which to test whether, how much, and how well a student stays inside or outside the bounds of the envelope, and the assessments (e.g., pass/fail) to be made on measurement values. For example, students might be assessed in terms of whether they were in the appropriate ‘proximity zone’ when they engaged an enemy aircraft, where the proximity zones are defined in an ME. Conversely, the student might be assessed on whether they failed to operate within the ME. Such violations of the envelope are sometimes called ‘exceedances’. In some domains, such as Military Flight Operations Quality Assurance, they are called ‘Events’; this has a pejorative connotation. An example of an Event might be when a helicopter banks dangerously, at an angle exceeding 60 degrees. This is equivalent to saying that the helicopter went outside of an ME that describes (and implicitly proscribes) bank angles that are less than 60 degrees.

MEs can provide a simple way to express what would otherwise require a lengthy enumeration of specific scenario conditions (as in Woods et al. 2015.) For example, an ME might specify that a tank should move within the bounds of flat areas that have hills with slopes less than 10 degrees and are not swampy. This ME can apply to a wide variety of regions in the terrain — the region should contain relatively flat areas and may or may not contain swampy areas — so the ME can be applied to it to assess the tank’s movement in all the regions with these properties. This could be done with scripts that define the solution paths precisely, but only with difficulty because there will be many different paths in many different regions.

Assessments generally describe ranges of performance within an ME, and may denote performance bands within the ME. For example, a pilot landing a plane with only small deviations from ideal glideslope may be in the expert performance band, but if the pilot’s deviations from ideal glideslope are larger (but still within the ME and therefore acceptable), the pilot may be in the novice performance band. Performance bands can be described by sub-constraints on envelope variables. In a spatio-temporal ME assessing adherence to navigation guidance and timelines, for example, a band within the ME that

![Diagram](image-url)

**Figure 7.** Detail for Training Objective Packages element. The TOP element in HPML contains a description of the conditions that constitute a training opportunity for a given TO together with the measures and assessments associated with the TO. In the current version of HPML, conditions are described verbally.
describes 1% deviation from the specified route might be assessed as expert performance, and a band that allows 5% deviation might be assessed as novice performance.

We note that the measurements and assessments in TOPs need not be MEs; MEs, however, provide a useful way to think about measurements and assessments that exploit the same underlying technology that is used to describe scenario conditions, namely Bes; so their use will often be natural in TOPs.

**Examples of MEs**

A few examples of MEs illustrate their utility:

- For a TO related to helicopter buddy lasing (mentioned in a TOP example, earlier), a valid shot ME might involve a specification that the helicopters fly within a certain range of the target (not too close and not too far), that their geometry with respect to each other and the target be within certain parameters, and that the rocket be fired within a certain interval after the laser acquired the target. None of these variables — range, angle, or timing — would be defined as a specific value, but rather as a range of values, and the measurement itself would be whether the helicopters were within the ME, that is, whether or not the shot was a valid one. Finer grained assessments of performance could come from performance bands within the ME, as explained earlier.

- In an air strike scenario in which student pilots are harassed by enemy aircraft as they approach their target, the students may respond to the enemy. However, the TO specifies that pilots must not fall too far behind on their timeline. This can be measured with an ME expressing how far ahead of or behind the expected timeline position the student can be, and assessments can measure both deviation from ideal timeline and the completeness with which students deal with the unidentified, potentially hostile aircraft.

- Pilots can perform certain manoeuvres to avoid being caught directly in enemy radar. An envelope that describes the spatio-temporal boundaries available to the pilot in which the enemy radar is the weakest can be easily described using envelope technologies, and a performance measure for the pilot is the extent to which he/she is able to stay in that envelope.

- In a serious game focused on training teamwork in an operating room or a critical care ward, the student may have considerable freedom of action. A given TO might be realised in many team tasks within that environment, or in many interactions. There may be many invalid interactions, such as reporting a significant change in patient state inaccurately, or failing to report it at all. MEs can formally specify these, and the associated assessment envelopes can assign categories of medical expertise: ‘Novice, Advanced Beginner, Competent, Proficient, Expert’ (Benner 1984).

**Benefits of MEs**

MEs benefit training designers during planning and they aid instructors and students during training execution and assessment. They do so because of several key features:

- MEs capture important spatio-temporal (and other) constraints on desired student behaviour.
- MEs provide the flexibility to measure only important variables and to omit those of no importance.
MEs can conveniently express measurements and assessments that would otherwise require the specification of a large, potentially inestimable, set of student behaviours in the scenario.

MEs provide an analogue to other constraint-based approaches to measurement, as described earlier.

Applications and research

While TOPs themselves are a relative newcomer to the scenario-based training world, several precursors have been, or currently are, in development. For example, Stacy, Walwaniis, and Colonna-Romano (2007) describes a constraint-based system for planning and replanning scenarios on the basis of TOs; Stacy, Colonna-Romano, and Roberts (2009) describe an envelope-based system for creating scenarios from flight logs; Wray and Woods (2016) and Woods et al. (2015) describe a constraint-based method to capture expert responses for improved student performance measurement in complex training environments; Jones et al. (2015) describe an envelope-based system for validating the behaviour of computer-generated forces; and the authors and their colleagues have work underway that uses a BE-based system to create realistic maritime patterns of life from real-world raw data.

In the future, we envision several potential lines of research that may enhance the impact of scenario-based training. First, the use of TOPs in the planning of a scenario-based exercise, especially a complex one, could dramatically simplify the planning task and dramatically increase the amount of training available to participants within the exercise. Instructors will specify TOs, the constraints that constitute conditions for those TOs, and their preferred measurements and assessments relevant to those TOs; the plannable TOPs will then be used as a main component of the exercise plan. We believe that there are other considerations such as a story and MSEL elements, scenario coherence constraints, and training and safety rules, which will also be needed; combining them with TOPs will lead to a very powerful plan. Further, some TOPs, such as the student’s ability to recover from certain kinds of mistakes, will not be suitable for planning, since it is difficult to plan for a student mistake. If these unplanned TOPs are known before the scenario begins, however, the scenario can be monitored for their conditions, and the instructor can be notified when those conditions occur.

Another possibility will be to provide instructors with a TO scoreboard for a scenario, both during the exercise and afterwards. A similar strategy has been implemented for performance measures (Wiese et al. 2012), and we believe it would be valuable to extend that approach with TOPs. The scoreboard could easily be used as a quick reference for instructors for AARs and even During-Action Reviews (DARs), and it could also help instructors keep track of what intended TOs were not addressed in the exercise.

TOPs will play an interesting role for certain kinds of scenario-based adaptive training. TOPs can be used to structure an exercise analogously to what vanLehn (2006) calls the outer loop of intelligent tutoring systems, where the TOP corresponds to the high-level, complex, multistep task. When a student has completed work on a TOP, the next TOP can be adaptively selected, based on students’ performance as measured during execution of the TOPs they have already encountered. This can be done in a variety of ways, including the use of advanced mathematical models like partially observable Markov decision
processes (Freeman, et al. 2009; Levchuk, Shebilske, and Freeman, 2012), which will bring benefits such as providing the optimal time-on-task for each student and optimising both student readiness and the time required to achieve that readiness.

The mechanisms defined here could eventually support distributed training control, especially in large LVC exercises, discussed briefly in the introduction. These events can be so complex that a central planning and control function is not feasible, or they may be physically distributed across unreliable networks. Under these circumstances, it will not be possible, from a central location, to monitor scenario conditions or to replan TOPs that have not yet been addressed. Since BEs are constraint-based, these network monitoring and replanning challenges can be cast as distributed constraint satisfaction problems in which only network neighbours can communicate, not everyone. Fortunately, solutions to distributed constraint satisfaction problems are emerging (cf. Modi et al. 2005; Faltings 2006), and an LVC exercise with disconnected groups of participants will be able to leverage them.

Finally, there are opportunities to extend TOPs to represent additional training constructs, such as instructional strategies. Instructional strategies specify how instruction will be delivered to fulfil TOs, including the best way to scaffold the student’s experience, and perhaps measures to identify when such scaffolding should be made available. The strategy may be fixed, defining one sequence of training objectives, and one method for providing feedback. It might be adaptive, in which case it might specify the training conditions or measured student state that trigger a selection among strategies, and that determine the choice. For example, an adaptive strategy might specify that students who are assessed to be novices receive immediate feedback when they err on a task step, and that more proficient students receive feedback only when all task steps are completed, per Shute (2008). Another strategy might specify the timing of feedback differently for each of several training conditions that support a given TO; immediately for conditions that allow the student to execute each task step deliberately (such as planning a course of action), and delayed for conditions that require the student to move briskly, without pause, within and between steps (such as a battle against many enemies). Yet another form of adaptive instructional strategy might condition the selection of training conditions on the history of training conditions experienced by the student. This would enable a TOP to develop in students a deep but narrow expertise through repeated administration of similar training vignettes, or generalised skill through administration of many, strongly varied vignettes. These examples are just a few among many, but they illustrate the value of packaging training objectives with the strategies for addressing and achieving them given certain conditions and measures.

Conclusion

TOPs answer the challenge articulated in the introduction to this article: to create experiential training that is reliably and measurably focused on training objectives. TOPs do this by specifying TOs, scenario conditions, and performance measurements and assessments in a way that allows both human training professionals and automated scenario infrastructure to understand, monitor, and manipulate them. This improves the management of complex exercises, accommodates unexpected behaviour from students or from computer-generated entities, and even identifies unplanned training opportunities.
The introduction of TOPs to experiential training events need not be disruptive. The first step will involve, prior to the event, elicitation of intended TOs, their required scenario conditions, and their associated measures. This is the best practice in instructional design. The use of TOPs codifies these elements of the training plan in human- and machine-readable form, which is an advance on best practice. During training, exercise managers could, in the simplest of cases, use this information for manual monitoring of TOs. However, it will be straightforward — and desirable — to develop automated infrastructure to monitor scenario conditions and performance measures. This will free up training personnel and can automatically record each student’s TO-relevant experience and performance.

A more advanced application will develop and apply automated infrastructure during scenario planning and execution. Because TO conditions can be quickly arrayed in a schedule, this holds the promise of significantly shortening the initial planning cycle. Further, should the exercise itself unfold in unpredictable ways, it will be feasible to do on-the-fly replanning to recoup as many training opportunities as possible for each student, resulting in a new schedule that can be implemented by revised instructions to the students and by run-time commands to the computer-generated entities. Eventually, the commands to the computer-generated entities can themselves be automated, and this ‘outer loop’ approach can work in conjunction with ‘inner loop’ adaptations (vanLehn 2006) that tailor each TOP-related vignette to individual student requirements as training unfolds.

Finally, TOPs will transform the third phase of the training process — assessment — by ensuring that measurement is rigorous at multiple levels. Measurements of student performance will populate feedback to them, and will help instructors to diagnose performance failures and the unexpected successes that denote learning or luck. Measurements of TO completions will enable instructors to assess whether a specific training covered the intended ground, and it will help their institutions to determine which TOs a curriculum reliably achieves, which it does not, and which training events are responsible for those effects. These assessment data will enable researchers to perform new forms of educational data mining that identify ways to refactor (or restructure) training and measurement to achieve new efficiencies.

TOPs hold the promise of increasing the impact of all kinds of training events, especially large, resource-intensive ones such as military training exercises. They provide principled, systematic, and strong support for planning, execution, and assessment components of training. They aid students by focusing training time on the intended TOs, and reducing the time wasted in activities that do not train skills up (and may even train them down). TOPs benefit instructors by freeing them to help students achieve TOs rather than on monitoring and managing the scenario. They aid instructional institutions by making training designs explicit and automatable, so that training missions are achieved efficiently and accountably.

**Acknowledgments**

We would like to thank Melissa Walwanis, Beth Atkinson, Ami Bolton, Heather Priest, Jeff Grubb, Brent Olde, John Colonna-Romano, Bob Wray, Randy Jones, and J.T. Folsom-Kovarik for thoughtful discussions about behaviour envelopes and their relation to Training Objective Packages. We
also thank Dylan Schmorrow, Sae Schatz, and other anonymous reviewers for their useful and insightful comments concerning this article. The authors of this paper are employed by and are stockholders in Aptima, Inc., a company that performs research and development for the U.S. Department of Defense.

**Disclosure statement**

No potential conflict of interest was reported by the authors.

**Funding**

The research reported here was partially supported by contracts N68335-12-C-0146 and N00014-12-G-0546 from the US Navy.

**About the authors**

Webb Stacy, PhD, corporate fellow at Aptima, focuses on using modern cognitive science to improve experiential training. His recent work includes using behaviour envelopes to create realistic maritime patterns of life and to optimise the training value of experiential scenarios. Dr Stacy holds a PhD degree in Cognitive Science from the University of Buffalo.

Jared Freeman, PhD, chief scientist at Aptima, has served as principal investigator in many research and development projects to analyse, model, train, and aid operators in military and medical missions. Dr Freeman received a PhD degree in Human Cognition and Learning from Columbia University.

**References**


