Using Pedagogical Information to Provide More Effective Scenarios

Webb Stacy, Ph.D.[1], Melissa Walwanis Nelson[2], John Colonna-Romano[1]

[1]Aptima, Inc.

Woburn, MA
wstacy@aptima.com
jcromano@aptima.com

[2]Naval Air Warfare Center Training Systems Division

Orlando, FL
melissa.walwanis@navy.mil

ABSTRACT

Modern simulation environments provide powerful practice opportunities for warfighters. Current approaches to scenario definition in these environments specify terrain, platforms, and major planned events, but link only indirectly to training goals. To move these environments from effective practice to effective training, they must incorporate pedagogical knowledge such as training objectives, performance measures, and trainee feedback. We have been working on an approach that provides guidance for scenario design, execution, and review. The approach views scenarios as collections of potentially overlapping learning episodes structured by a construct called the *experience*—a specific scenario-based situation that will give trainees an opportunity to make progress towards their training objectives.

The conditions that will bring about a particular experience are expressed as constraints. In practice, these are straightforward statements of conditions in the scenario that must be true in order for the learning episode to take place. For example, in order for a helicopter pilot to work on a training objective in sensor fusion, multiple sensors must be enabled in the helicopter, the targets of interest must be within their range, and it must not be raining. Normally, this is accomplished during scenario planning by placing an event on a Master Scenario Events List (MSEL) that calls for relevant platforms to be in specified places at specified times. But those specifics are often unnecessary from a training perspective. Relaxing them—while enforcing the experience-based constraints—provides the possibility of additional learning opportunities when the scenario does not unfold exactly as expected.

The resulting scenario provides trainees with an environment that lets them make progress more reliably against their training objectives, and this results in more effective training. This paper explains the approach and illustrates its use in a military scenario-based training environment.

ABOUT THE AUTHORS

Webb Stacy, Ph.D., is Vice President of Technology at Aptima. He oversees Aptima's current and future technology portfolios. His focus is the intersection of software and computer science with the science, modeling, and measurement of warfighters as individuals and as teams. He has extensive experience in the development of mission critical software, and holds a Ph.D. in Cognitive Science from the State University of New York at Buffalo and a B.A. in Psychology from the University of Michigan.

Melissa M. Walwanis Nelson is a Senior Research Psychologist at the Naval Air Warfare Center Training Systems Division in Orlando, Florida. She has a M.S. degree in Industrial/Organizational Psychology from the University of Central Florida. She manages a program of research devoted to exploring instructor station enhancements and is the lead for the development of the Common Distributed Mission Training Station. Her research interests include distributed mission training, leadership development, simulator instructional tools, and network centric warfare concepts for coalitions.

John ("JCR") Colonna-Romano is the Lead Software Architect at Aptima, Inc. Mr. Colonna-Romano has over 25 years of experience in software system architecture, systems engineering and software engineering. He is interested in system architecture, simulation based training systems, computer game technologies and the application of technology to solve complex problems. Mr. Colonna-Romano has co-authored two books on distributed systems middleware architecture. He started his career at the U. S. Department of Defense. Additionally, he earned a B.S. and an M.S in Computer Science from Stevens Institute of Technology.

Using Pedagogical Information to Provide More Effective Scenarios

Webb Stacy, Ph.D.[1], Melissa Walwanis Nelson[2], John Colonna-Romano[1]

[1]Aptima, Inc.

Woburn, MA

wstacy@aptima.com

[2]Naval Air Warfare Center
Training Systems Division

Orlando, FL

melissa.walwanis@navy.mil

THE SCENARIO DESIGN CHALLENGE

In a distributed, simulation-based training exercise, the opportunity for a helicopter pilot to practice using sensor fusion to classify contacts is lost when there are no targets in range at the scheduled time. The crew of a command and control aircraft, meanwhile, helps coordinate important aspects of the mission, but has no opportunity to receive training of their own—they are serving as little more than a training aid. Elsewhere in the exercise, fighter pilots miss the chance to develop their air-to-air skills when the planned surprise encounter with enemy aircraft is missed because the pilots were ahead of the timeline.

Distributed simulation-based exercises provide trainees with opportunities to develop knowledge and skills, but, as these examples illustrate, many additional opportunities are missed when unexpected events cause deviations from the initial training plan. Because these exercises involve the time and effort of many participants, a lot of work goes into planning scenarios. Typically, participant organizations determine what they need their trainees and teams to get out of the exercise and then attempt to accommodate all of those objectives in the scenario. It is remarkable that such a complex planning activity results in such typically coherent plans.

Unfortunately, just as battle plans do not survive first contact with the enemy, exercise plans rarely survive first contact with the participants. They are, after all, trainees. They will fly, drive, and otherwise operate their virtual simulators as best they can, but inevitably the fact that they are acquiring new knowledge and skills means that their virtual behavior will cause the exercise to deviate from the original intent. The fact that the exercises are often geographically distributed amplifies this problem because communicating any required scenario adjustments becomes more complicated than making a simple request of a person sitting in the same room. Add to this the potential difficulties with equipment, communications failure, and the overhead associated with coordination of human support personnel, and we conclude that the intended training of large scale exercises differs from the reality.

One result is that the exercises do not present trainees with as many robust training opportunities as they could. This paper discusses research that addresses ways that some or all of these opportunities might be recovered. The fundamental idea is that supplementing typical scenario information with information about training objectives (TOs) and the conditions necessary for trainees to work towards them, and using the supplemental information appropriately during planning and during the exercise, will dramatically increase the effectiveness of training exercises and therefore increase operational readiness.

THE PEDAGOGICAL ARCHITECTURE

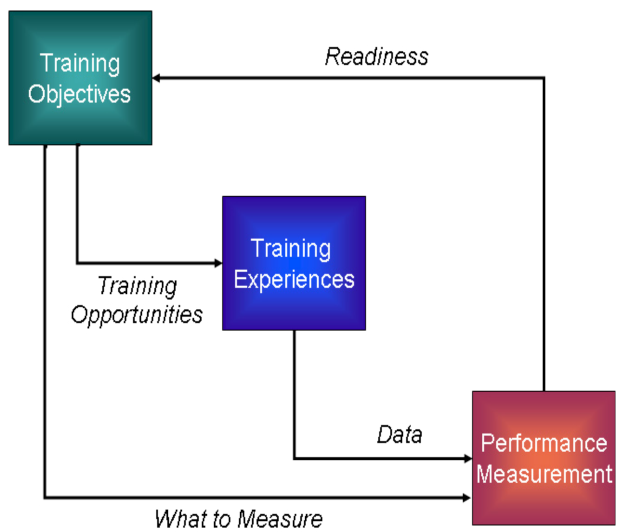

Figure 1. Pedagogical Architecture

Figure 1 shows the major ingredients of a pedagogical architecture for scenario-based training. It was developed as a part of the Defense Advanced Research Project Agency's DARWARS program, and it is described in more detail in Weil, Hussain, Brunye, Sidman, & Spahr (20005). In this architecture, trainees have specific *TOs* to gain competencies, knowledge, or skills in specific subjects or domain areas. These objectives are then matched to the appropriate *training experiences* in which they make progress against the TOs. Because trainee responses within the simulation are recorded and contextualized in terms of the TOs (Stacy, Merket, Puglisi, & Haimson, 2006), *meaningful measures* can be made based on trainee performance, recorded, and then extracted to provide assessment and feedback to trainees (Stacy, Ayers, Freeman, & Haimson, 2006; Stacy, Merket, Freeman, Wiese, & Jackson, 2005). The measures give trainers insight into trainee progress with regard to the objectives.

All three components of the architecture will improve training efficacy in scenario-based exercises, but the key component, from the point of view of increasing the number of opportunities for training in the scenario, is the Training Experience. Training experiences themselves have two components, as illustrated in Figure 2. They are enabled by conditions in the scenario necessary to work on the TO, or TO Conditions. For example, in order to work on TOs relating to air-to-air combat, there must be an airborne enemy to engage. Once the TO conditions are met, the trainee engages in that work, the Training Activity. In the air-to-air example, the training activity consists of the trainee actually performing the tasks relating to air-to-air combat. When the trainee performs these tasks, data are generated, either from the system, from an observer, or in some cases from the trainees themselves, and these data can be used as a basis for measurement and assessment of trainee performance.

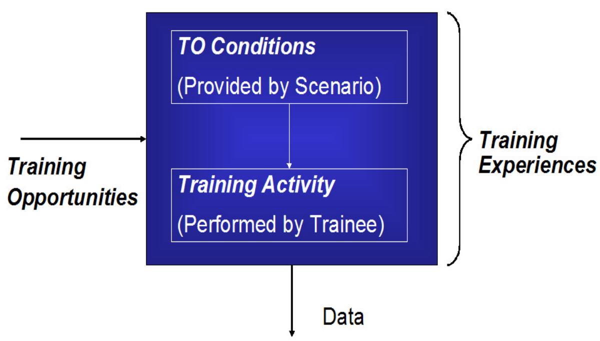

Figure 2. Experience-based Scenario Conditions

In this paper, we focus primarily on bringing about the appropriate TO conditions in the scenario. These conditions are sometimes obvious. For example, contacts must be in range in order to address a TO concerning sensor fusion for classifying contacts. Other times, the conditions are less obvious. Certain sensors require certain kinds of weather, for instance, and certain evasive maneuvers require certain kinds of enemy radar in certain postures. But regardless of whether the conditions are obvious or not, the complexity of many TOs for a diverse training audience means that without some kind of explicit representation of TO conditions and automated support for creating them, those opportunities will be lost.

TOWARDS PEDAGOGICALLY ORIENTED SCENARIOS

The explicit use of the pedagogical content of training scenarios before, during, and after the simulation training exercise will increase the number and quality of training opportunities. Specifically, the formal capture and representation of the TO condition requirements of a mission enables automated systems to optimize training feasibility before the exercise and to recommend useful changes to the scenario during the exercise. We will discuss how this is done in the next section.

Figure 3 shows a notional diagram of the effects of this optimization. It shows the probability of the trainees achieving their TOs (in the vertical axis) as a function of the overall complexity of the training. This complexity has several dimensions such as the number of trainees, the diversity of TOs, and the experiential fidelity of the environment.

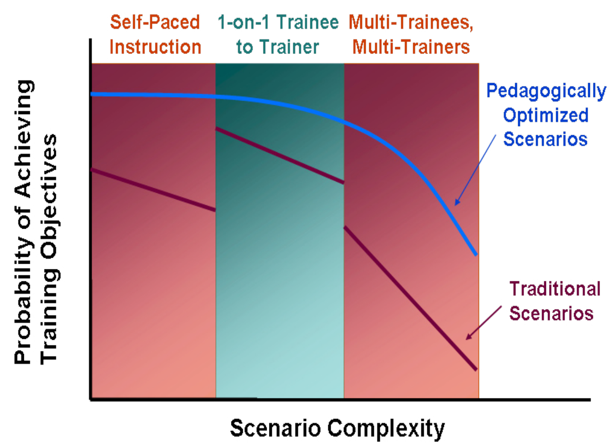

Figure 3. The Value of Using Pedagogical Information in Simulator-based Training (Notional)

Pedagogically optimized scenarios can play a significant role in at least two places on this complexity continuum. First, in those circumstances where a trainee will be using a standalone PC-based simulation for self-paced instruction, without an instructor, optimized scenarios can ensure that the simulation delivers the right conditions for the trainees to make progress against their TOs.

There is a time—typically before complex scenario-based training but after the student has gone through substantial classroom instruction—that trainees are trained individually in simulators, with instructors controlling their experiences, as depicted in the middle of Figure 3. In some cases, the presence of the instructor and the instructor's expertise is sufficient to create the right TO-based conditions for the trainee. During this time, instructors may have grade cards to guide the sequence of experiences to be inserted by the instructor. The instructor's expert attention to the needs of the trainee reduces the need for automated support during this time, though there are certainly circumstances for which it could be valuable such as a shortage of instructors or expertise, or the presence of complicated devices that make it difficult for instructors to make changes. In larger exercises, when there are multiple trainees and multiple trainers, the probability of achieving TOs for both the traditional and the optimized methods begin to decline because of the complexity of the scenarios and the coordination and dependencies between trainees and instructors. Pedagogically optimized scenarios with automated support maintain a higher probability of achieving any given TO because the automated support can manage the complexity and dependencies across all trainees.

For example, consider a simulation-based training exercise with two trainees who have TOs where the second trainee's training opportunity is dependent on the first trainee performing a certain maneuver. If the first trainee fails to perform the required maneuver, then the second trainee will miss the training opportunity. A pedagogically-informed automated support system can help the first trainee by manipulating the simulation environment, for example by moving a Semi-Automated Forces (SAF)-generated object to a better place, so that both trainees will have their training opportunities.

Pedagogically optimized scenarios offer these additional benefits:

- They improve training consistency. Because the TOs and other pedagogical information are captured in the scenario definition, each time the scenario is used, the trainees can get a consistent training activity. This provides consistency across instructors, units and time.

- They reduce instructor workload. During a training exercise the instructor must monitor the trainee activity, pay attention to the scenario flow, keep in mind TOs and potentially interact directly with the databases underlying the simulation in real time. As the complexity of the scenario increases, this workload also increases, requiring either additional trainers or skilled operators.

- They support experiential variation. Pedagogically optimized scenarios describe only the conditions necessary for the training experience. These conditions are less restrictive than the timeline based sequencing of events in a traditional scenario definition, allowing more flexibility in the training experience while maintaining the essential conditions required. The resulting variation not only provides a broader experience for the trainee, but also limits the trainee's ability to know the exact scenario ahead of time and "game" the training exercise.

CREATING TRAINING OBJECTIVE CONDITIONS USING SCENARIO CONSTRAINTS

Clearly there are benefits to representing and using information about TOs and their necessary conditions. But as a practical matter, how do we do this? Our answer is to consider TO conditions as constraints on the scenario. Although this adds certain restrictions, it removes many others. The net result is a definition of the scenario that has more degrees of freedom and is better able to maintain a focus on TOs in the face of unexpected behavior from trainees or elsewhere.

To understand how this works, it is necessary to define a few simple concepts, adapted from Apt & Wallace (2006). Formally, a constraint is simply an expression involving one or more variables that could be true or false, for example

Ship1.displacement = Ship2.displacement
Time (Launch) precedes Time (Rendezvous)
Distance (Aircraft1, Aircraft2) < 5

When the constraints are true, they are said to be *satisfied*. *Constraint satisfaction problems* are ones whose solution involves finding values of the variables for which all the constraints are satisfied. When there is more than one solution, the problem is said to be *underconstrained*; when there is no solution, the problem is said to be *overconstrained*.

In a sense, scenarios are already planned using constraints. For example, an important aspect of existing scenario definitions is the list of planned events, often called a Master Scenario Event List (MSEL). Each event generally has a time, a location, a set of entities involved, and a synopsis of the expected actions and responses by the participants. We can therefore think of each event on the list as a constraint. For example, if an event on the MSEL reads like this

T+12 First helo joins second in tactical formation.

we can formulate this equivalently as a constraint

T = 12
AND
Distance (Helo1, Helo2) = tac_form_distance
AND
Rel_posn(Helo1, Helo2) = tac_form_rel_posn

Having expressed the event this way, we can feed it (and the other events on the MSEL) to a constraint satisfaction application to discover where each entity should be, and when, in order to meet the MSEL.

Very likely the MSEL is already committed to a specific location and time for each event. It is possible to find these specific locations and times manually because the problem of accommodating events in the MSEL is greatly under-constrained, that is, there are many acceptable solutions. As a result, the specific locations and times in the MSEL are often stricter than necessary, as we discuss shortly. To the extent it is possible to relax the implicit time and space constraints in the MSEL, variations of the scenario will be automatically enabled on subsequent execution, since MSEL events will not always happen at the same time or place. This will provide built-in experiential variation for trainees that encounter the scenario more than once. In any case, even if we leave the MSEL untouched we may be able to accommodate more conditions for more TOs if we feed those additional conditions to the constraint satisfaction application, and the same information can also help to recommend beneficial changes during the exercise when the scenario doesn't go according to plan.

In general, it will be important to represent conditions for TOs in addition to those in the MSEL. To do so, we need to be able to express three kinds of constraints: time-based constraints, space-based constraints, and other kinds of constraints.

Time Based Constraints

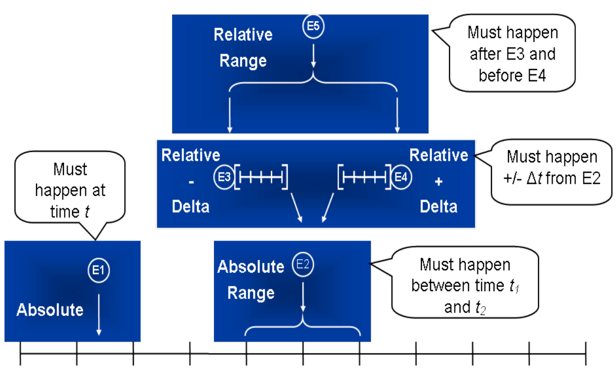

Figure 4. Time Based Constraints

Part of the value of using constraints to characterize scenario conditions is the flexibility of defining only the necessary constraints and allowing other aspects of the scenario to vary. In a timeline based scenario definition, the time constraints are defined as absolute points in time. In some cases this is necessary, but in many cases an absolute time constrains the event more than necessary. A more flexible way is illustrated in Figure 4. Specifically

- Absolute – This type of time constraint is used to tie an event to a specific point in time, as in the MSEL.
- Absolute Range – This type of time constraint is used to describe that an event must take place during a range of time. The event should not take place before or after this range.
- Relative – This type of time constraint ties an event to the occurrence of some other event. In this case, if the anchoring event is delayed, then the dependant event is also delayed, but it is the relationship between them that is the critical part of the constraint. These types of constraints are used to define the relative sequencing of experiences in the scenario.
- Relative Range – This type of time constraint defines a range in which the event must occur. This type of constraint (as well as the relative one above) will flex given the actual conditions in the scenario. If events are delayed or accelerated, the important conditions (defined as constraints) might still be met, and the training experiences delivered.

Space Based Constraints

Another important kind of constraints are those related to space and location. Space constraints can be defined as absolute locations, but this unnecessarily reduces the flexibility of delivering training opportunities. It may not be necessary for a trainee to rendezvous with a SAF generated aircraft at a pre-determined location, but rather to rendezvous within a certain area. Figure 5 shows a space constraint of the distance between the trainee aircraft and a potential target. The figure shows a range around the trainee where the target should lie in order to satisfy the distance constraint: Target-1 is within the range and Target-2 is not.

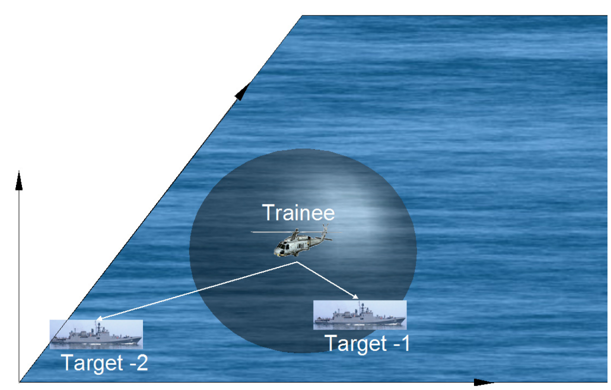

Figure 5. Absolute Distance Based Constraints

Line distance between a trainee and a target, or the trainee and some other element of the training environment may not be descriptive enough to fully describe the space constraints that must be satisfied. These constraints could apply to just one dimension of the distance, as well. A good example is that there may be TOs for which it is important that an aircraft be at a certain altitude.

Spatial constraints can be relative, too. Figure 6 shows a relative distance and direction constraint. In this case the two helicopters are engaged in the experience of flying in formation. Flying in formation requires a certain configuration of the aircraft. If one of the helicopters were SAF generated it would have to follow the given relative space and direction constraints to achieve and maintain formation with the trainee.

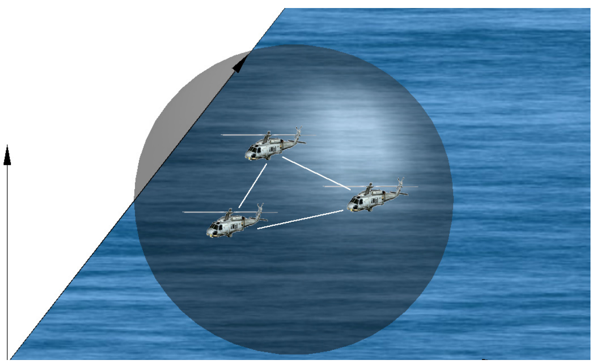

Figure 6. Relative Distance and Direction Constraints

It is worth mentioning that absolute and relative airspeed are constraints that involve both space and time.

Other Constraints

Constraints besides those describing space and time are important for describing conditions for TOs. For example, some training objectives can be met only when aircraft are over water, and others can only be met when the weather is clear. This class of constraints is harder to characterize than space and time, but will be no less important. The particular constraint technologies we have been working with are quite flexible in their ability to express these constraints, so it is straightforward to make them an integral part of the planning and execution of scenarios.

AN EXAMPLE SCENARIO

An example will illustrate these concepts. We have been working with a use case from a Navy helicopter community. The scenario involves one trainee and several SAF entities. The use case actors are:

- Computer-generated friendly ship S1 (MOM).
- Computer-generated friendly ship S2.
- H1 – Virtual helicopter simulator flown by pilot trainee. Originates from S1.
- H2 – Computer-generated helicopter originating from S2.
- Computer generated target ship T1
- Computer generated target ship T2

TOs for the example can be seen in Table 1.

Table 1. Training Objectives for the Example

High-Level Training Objectives	Subordinate Training Objectives
Fly in Formation	– Navigate to Rendezvous location – Maintain Safe Distance in Formation Flight

Classify Contacts	– Deploy sensors
	  – Deploy Sensor 1
	  – Deploy Sensor 2
	  – Sensor Fusion
	– Communicate with MOM

Attack Targets with Missile	– Deploy Missile – Buddy Lasing

Figure 7 shows the general layout of the scenario and the sequence of events from rendezvous and formation to arrival at the area of operation, sensor deployment, missile deployment and buddy lasing.

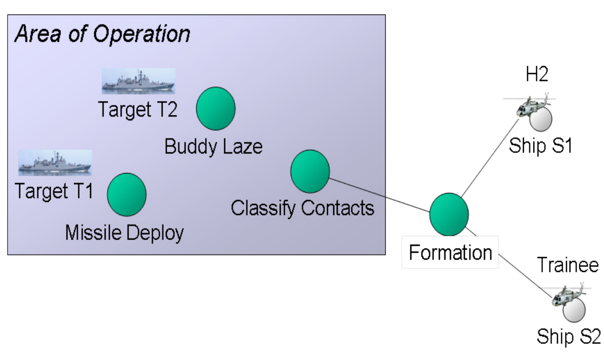

Figure 7. Scenario Map

Setting Up the Constraints

Including the TO conditions in the scenario makes explicit the constraints necessary to bring about the desired training opportunities. To understand how this works, we start with the constraints that describe a fixed MSEL-based scenario (Table 2) and then loosen them for increased flexibility.

Table 2. MSEL for Example Scenario

Time (h+m)	Event
0+0	Launch H1 from S1 (Trainee)
0+5	Launch H2 from S2
0+10	H1 rendezvous with H2
0+20	Arrive at Area of Operation
0+25	T1 and T2 enter Area of Operation
0+40	T1 threatens H1
0+50	T2 threatens H1
1+05	Return to Base

As an example, Figure 8 shows the conditions necessary for the TO called "Classify Contacts" as fixed, time-based constraints. The three subordinate TOs (and their conditions) are fixed at absolute points in time on the scenario timeline: at the first specific time, the trainee will work on the TO concerning Sensor 1, at the next on the TO concerning Sensor 2, and at the third on Sensor Fusion. Finishing work on the Sensor Fusion TO finishes work on the TO for Classifying Contacts, at which point the trainee's performance on Classifying Contacts is measured.

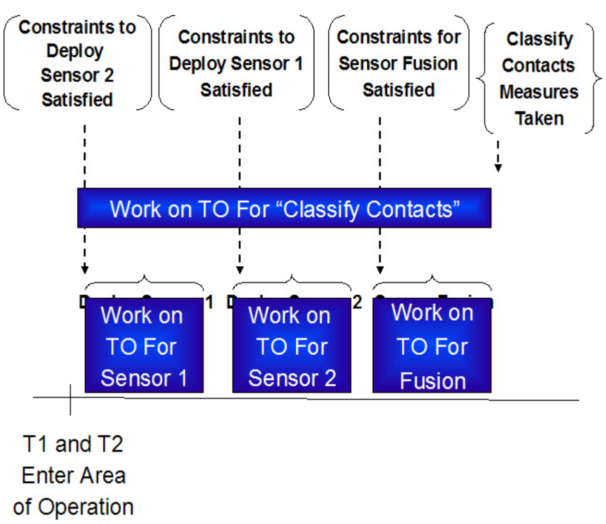

Figure 8. Explicit Constraint Timeline for Classify Contacts

Figure 9 illustrates the difference when strict timing and sequencing constraints can be relaxed. The upper section of the figure shows the fixed sequencing of the TO condition, but now the exact timing is omitted. The result is a graph showing the dependencies among the TO conditions.

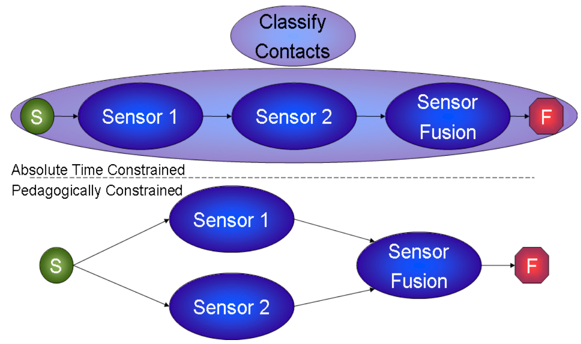

Figure 9. Sequencing Constraints for Classify Contacts

For example, work on Sensor 1 must come before work on Sensor 2, and both must come before work on Sensor Fusion, but the exact timing is left unspecified and therefore is free to vary with other requirements. In the lower section, one of those dependencies has also been relaxed—now, the order in which work proceeds on Sensor 1 and Sensor 2 is unimportant as long as both come before work on Sensor Fusion, providing even more flexibility to adjust to changing scenario conditions.

SCENARIO AUTHORING ENVIRONMENT

Because it is hard for humans to deal with more than a handful of constraints at once, the approach described here will be practical only if it is supported by automation. This section and the next describe considerations for implementing such automated support. The objective is to be able to assist both with scenario authoring, as described in this section, and with real-time monitoring and advice to SAF operators during the exercise, as described in the next section. We describe the scenario authoring aid as static analysis and the real-time exercise aid as dynamic analysis. In both cases, the goal is to maximize the training opportunities in an exercise by advising scenario designers, instructors, SAF operators, and even trainees on the steps they (or the system) can take to bring about additional TO conditions.

Figure 10 shows a notional user interface that expresses time and sequence constraints between TO conditions. The top panel shows the top level TOs of the exercise. There are buttons to add additional TOs which require explicit sequencing constraints. There are also buttons to add and remove the constraints. The constraints are represented as a network diagram. The center panel shows the expansion of the selected TO in the top panel. In the figure, the Classify Contacts TO is selected and the subordinate TO conditions that make up that TO are shown in the middle panel. The sequencing graphs are editable.

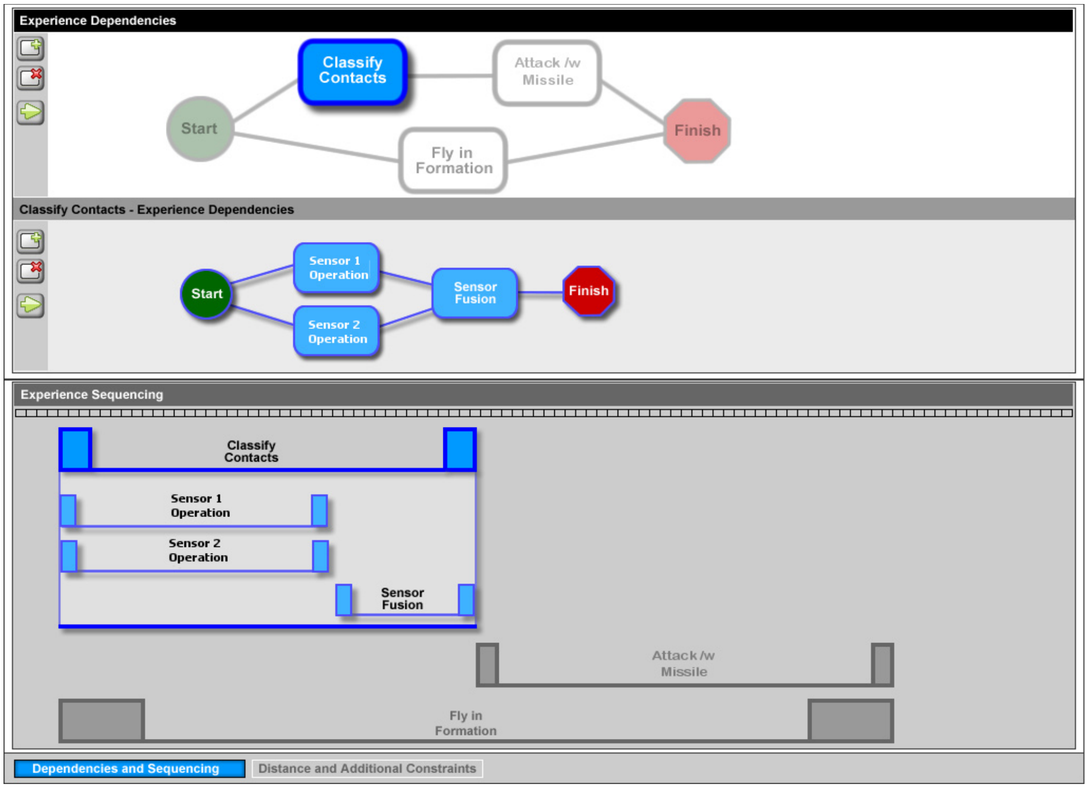

Figure 10. Defining Sequencing Constraints

The lower panel of Figure 10 shows a static scenario analysis view. The sequencing of the TO conditions in the scenario is presented as a scheduling problem. The lower panel of this figure shows a solution to the sequence constraints of the TO conditions as a Gantt chart. Each row in the Gantt chart shows a TO. The bounds of the row show the range of times that an experience could occur. This analysis isn't to actually schedule the experiences but to visualize the set of schedules for the scenario that will meet all of the constraints. If there is no valid scheduling of the experiences, then the system will provide visual suggestions about how to adjust the constraints so that there is at least one feasible schedule.

Figure 11 shows the conceptual user interface for expressing distance constraints for TO conditions.

The header of the window shows the TO for which conditions are being developed, namely Classify Contacts. The top panel shows distance constraints between the trainee and either the target or the sea surface. These are the result of specifying that a target must be in range of both Sensor 1 and Sensor 2 in a specific altitude range in order to perform sensor fusion for classifying contacts. The center panel shows other constraints such as weather state, and these are also related to conditions necessary to perform sensor fusion. The visual display in the lower portion of the screen graphically shows the ranges of the sensors. The operational range overlap shows the intersection of the two sensor ranges indicating the distance the trainee needs to be from the target in order to satisfy the sensor fusion TO constraints. The right side of the lower portion of the screen shows the initial positioning of the targets and the range constraints overlaid on them. This screen is used for scenario authoring to capture the constraints and help the scenario author analyze the scenario. A similar screen to the bottom section will also be used during the training exercise. In this case the targets, SAF entities and trainee will be dynamic and possibly moving, but the meaning will be the same—targets need to be in range of both sensors.

TRAINING ENVIRONMENT

The technology described above provides automated support for trainers and SAF operators to create the appropriate TO conditions during the exercise. As discussed, during scenario design the sequencing of TO conditions can be viewed as a scheduling problem that lends itself to Gantt charts to visualize solutions. As the training exercise progresses, the set of schedule solutions can change as unexpected events arise, and the new set of solutions can also be usefully presented as Gantt charts. Figure 12 shows two time points in an exercise and the dynamic scheduling solution for each. The top diagram shows an early point in the exercise where the constraints have not been met for

Figure 11. Defining Distance Constraints

any of the experiences. The red vertical line indicates the current time on the graph (the time line is shown at the top). The display indicates that the NavigateToRendezvous TO conditions can happen anytime between time 10 and 60 and the SensorFusion TO conditions can happen anytime between time 15 and 125. The length of the bar on either end of the range indicates the estimated time to accomplish the training activity. As time progresses and the red line moves to the right, the schedules will change to reflect the current state of the exercise. This information can be used by trainers or SAF operators during the exercise to bring about the planned TO conditions when unexpected events threaten to disrupt them.

The lower window in Figure 12 shows a time later in the exercise (time 60). At this point in the exercise, the DeploySensorl and DeploySensor2 experiences have been accomplished and are indicated as blue bars. These two experiences are represented as bars rather than ranges because the actual experiences have occurred. The figure also shows that the time range for the NavigateToRendezvous has passed. This indicates that there is no longer a solution to meet all of the TO conditions for this experience in this exercise. Finally, the window indicates that there are still possibilities for the ProperFormation, SensorFusion and CommunicateWithMOM TO conditions to happen. Not all the TO conditions have been met, or the trainee has not completed the training activities yet.

More detail on what needs to happen to create the conditions or complete the training activity is available to the user on request, as can be seen in Figure 13. Here, the SAF operator has requested more information about the Sensor Fusion TO conditions, and is being told that while Sensor 1 is active, Sensor 2 is not, and that there is not a suitable target in range. The instructor will ask the trainee to activate Sensor 2 and the SAF operator will move the target into the range of the sensors. When this is done, the trainee will be able to begin work on the Sensor Fusion TO.

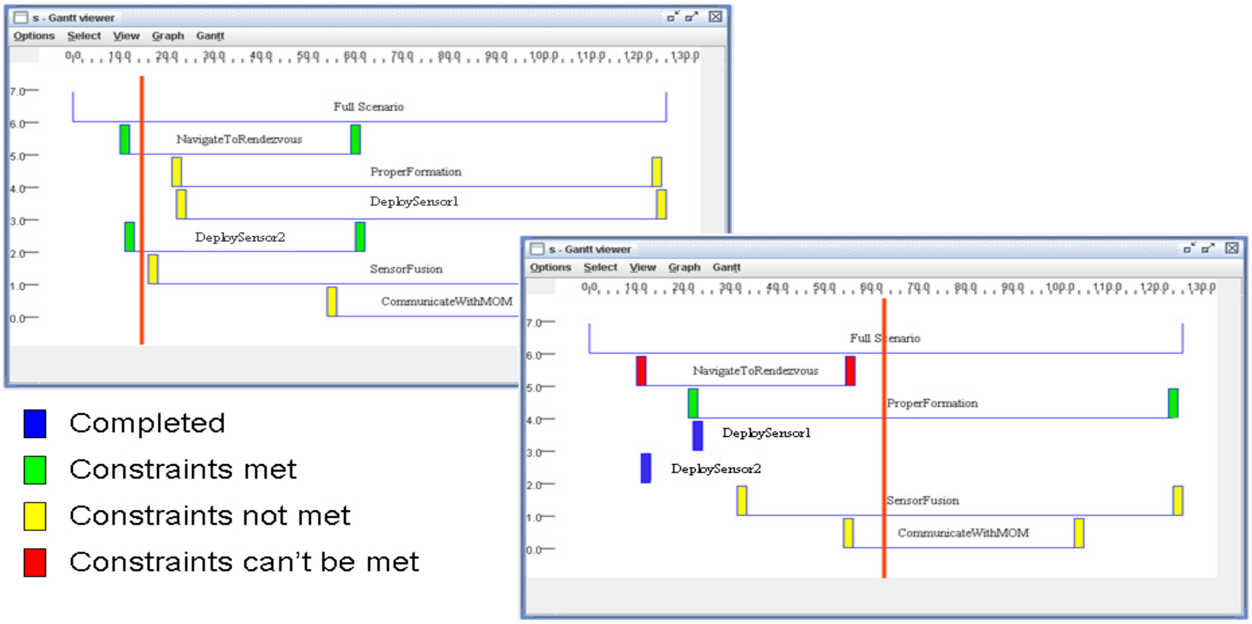

Figure 12. Gantt Experience Scheduling Views at Different Points in Exercise

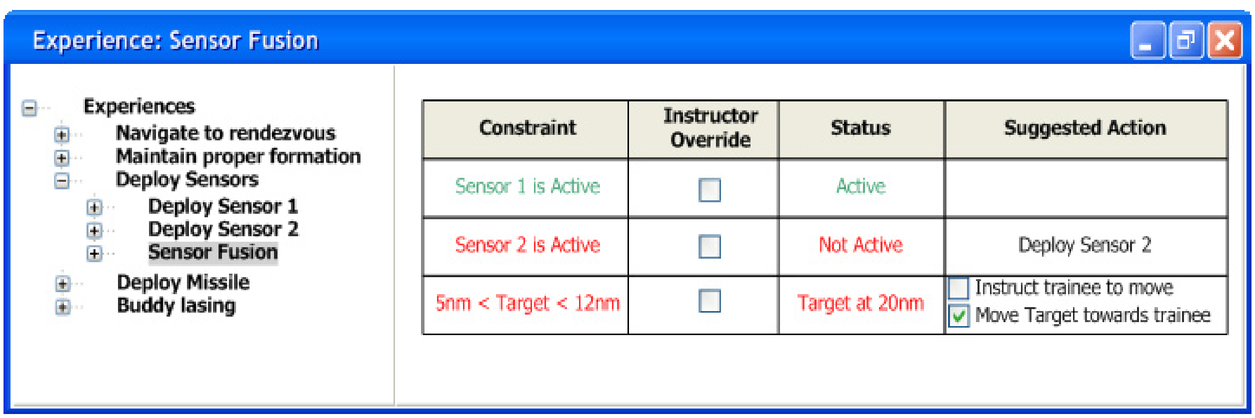

Figure 13. Detail on Additional Conditions Required to Complete an Experience

CONCLUSION

We have seen that pedagogical information—information about training objectives and their related scenario conditions, measurements, and assessments—can be valuable in the design and execution of simulation-based scenarios. In particular, expressing scenario TO conditions as constraints, and keeping designers, instructors, and SAF operators informed about solutions that satisfy the constraints, has these benefits:

- An increased probability that trainees will have meaningful training opportunities to make progress against their training objectives when the scenario doesn't go exactly as planned;
- An increase in experiential variation for the trainees, leading to a broader experience base and a decreased ability to game the system;
- A reduced workload for SAF operators and instructors, allowing them to focus on enhancing the trainees' learning experiences, tailoring training based on trainee performance, and managing and controlling additional SAFs; and
- Ultimately, providing increased force readiness at a reduced cost.

We will be implementing an automated system with the functions described above to meet the needs of the Naval Air Systems Command. We are implementing support for instructor specification of TO conditions, for optimizing initial scenario conditions, and for during-exercise monitoring. Our goal is to apply this approach to small portions of real exercises in order to refine the approach in directions that real trainers and trainees find most useful. We strongly believe that the approach will enhance the usefulness of distributed, simulation-based training exercises for all participants, and we are eager to see if the organizations who make heavy use of simulation-based training also make heavy use of this strategy.

ACKNOWLEDGEMENTS

The Navy Program Manager for Aviation 205 funded this R&D effort. It is managed by Naval Air Warfare Center Training Systems Division, Orlando, FL. The opinions expressed here are those of the authors and do not necessarily reflect the official position of the sponsors or the Department of Defense.

REFERENCES

Apt, K., & Wallace, M. G., (2006) *Constraint Logic Programming using Eclipse*, Cambridge University Press.

Stacy, E.W., Ayers, J., Freeman, J., & Haimson, C. (2006) Representing Human Performance with Human Performance Measurement Language. *Proceedings of the Fall 2006 Simulation Interoperability Workshop*, Orlando, FL.

Stacy, E.W., Merket, D., Puglisi, M., & Haimson, C. (2006) Representing Context in Simulator-Based Human Performance Measurement. *Proceedings of the 2006 Interservice/Industry Training, Simulation & Education Conference,* Orlando, FL.

Stacy, E.W., Merket, D., Freeman, J., Wiese, E., & Jackson, C. (2005) A Language for Rapidly Creating Performance Measures in Simulators. *Proceedings of the 2005 Interservice/Industry Training, Simulation & Education Conference,* Orlando, FL.

Weil, S. A., Hussain, T. S., Brunye, T., Sidman, J., & Spahr, L. (2005). *The Use of Massive Multi-Player Gaming Technology for Military Training: A Preliminary Evaluation.* Paper presented at the Human Factors and Ergonomics Society 49th Annual Meeting, Orlando, FL.