Creating Data Driven Training Scenarios

Webb Stacy, John Colonna-Romano,
Georgiy Levchuk, Brent Fegley
Aptima, Inc.
Woburn, MA and Orlando, FL
wstacy@aptima.com, jcromano@aptima.com,
georgiy@aptima.com, bfegley@aptima.com

J.T. Folsom-Kovarik,
Charles Newton
SoarTech, Inc.
Orlando, FL
jeremiah@soartech.com,
charles.newton@soartech.com

ABSTRACT

We live in a virtual explosion of data. The Internet generates an estimated 2.5 quintillion bytes of data every day. Though the data from instrumentation on aircraft, vehicles, ships, autonomous systems, simulators, and, increasingly, humans themselves does not reach this scale, its volume is significant and increasing. It is natural to want to use this wealth of data to build realistic training scenarios.

The chief difficulty is that, whatever events were recorded, they represent only one path through the world. This makes the recording suitable for replay, but a recording cannot give students the chance to make choices in the simulated world that would take them down different paths. Recordings cannot be directly used for training scenarios unless additional steps are taken. This means that accommodations must be made, through subject-matter expertise, machine learning, or both, to synthesize the data into realistic entity behaviors in a scenario.

In this paper, we discuss our experiences building several systems that take these additional steps, which generally involve machine learning and intelligent agents, and we discuss in detail an effort that focuses on creating realistic constructive maritime patterns of life from real-world data.

We conclude by discussing the training value of learning patterns of life from real world data, and lessons learned that will be useful to help other training professionals create realistic data-driven training scenarios.

ABOUT THE AUTHORS

Dr. Webb Stacy is a Corporate Fellow at Aptima, Inc., where he is responsible for enhancing Aptima's technology portfolio. Dr. Stacy has an interest in using modern Cognitive Science to improve experiential training. His recent work includes investigating the relationship of simulator fidelity to training effectiveness and developing an approach to optimizing the training value of experiential scenarios. Dr. Stacy holds a Ph.D. in Cognitive Science from SUNY/Buffalo, and a B.A. in Psychology from the University of Michigan, and has served as Program Co-Chair for the Society for Behavior Representation in Modeling and Simulation.

John Colonna-Romano is a Software Architect at Aptima Inc., where he works on big data cyber analytics technology. Mr. Colonna-Romano is interested in simulation based training systems, cyber analytics, big data /parallel computing, software architecture and software engineering process. He holds BS and MS degrees in Computer Science from Stevens Institute of Technology and a certificate in Software Architecture from the Software Engineering Institute at CMU.

Dr. Georgiy Levchuk provides expertise in mathematical modeling, optimization algorithm development, and software prototype implementation. He specializes in multi-objective optimization of mission-based organizations, including the design of command, control, and communication networks; agent-based command and control simulation; and optimized training design. Dr. Levchuk received his Ph.D. in Electrical Engineering from the University of Connecticut, specializing in the design of congruent, robust, and adaptive organizations. He received his B.S./M.S.M. in Mathematics with highest honors from the National Taras Shevchenko University of Kiev, Ukraine. Georgiy is a member of the Institute of Electrical and Electronics Engineers.

Dr. Brent Fegley is a Senior Research Engineer at Aptima, Inc., where he is engaged in quantitative research, statistical inference, machine learning, and signal processing. Dr. Fegley received his Ph.D. in Informatics and M.S in Library and Information Science from the University of Illinois at Urbana-Champaign. Additionally, he holds an M.A. in Musicology from the University of Michigan, Ann Arbor, and a bachelor's degree in Music Theory/History from West Chester University of Pennsylvania.

Dr. J.T. Folsom-Kovarik is a Lead Scientist at Soar Technology, Inc. His research helps advance training and decision support systems via data analytics and adaptation to individual users. He contributes novel methods, algorithms, and knowledge representations that let intelligent systems automate and support onerous or difficult tasks for instructors, instructional designers, and commanding officers. The intelligent systems can help reveal individual or massed user interactions with the system, understand underlying reasons for the observations, and reason about the contexts, goals, and tradeoffs required to best support each learner or operational user. Dr. Folsom-Kovarik earned a Ph.D. in computer science from the University of Central Florida.

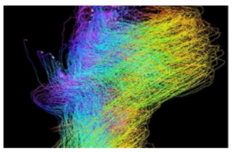
Charles Newton is an Artificial Intelligence Engineer at Soar Technology. His research and development experience covers human behavior modeling, natural language-based systems, and design patterns in intelligent systems.

Creating Data Driven Training Scenarios

Webb Stacy, John Colonna-Romano,
Georgiy Levchuk, Brent Fegley
Aptima, Inc.
Woburn, MA and Orlando, FL
wstacy@aptima.com, jcromano@aptima.com,
georgiy@aptima.com, bfegley@aptima.com

J.T. Folsom-Kovarik,
Charles Newton
SoarTech, Inc.
Orlando, FL
jeremiah@soartech.com,
charles.newton@soartech.com

INTRODUCTION



We live in a virtual explosion of data. The Internet generates an estimated 2.5 quintillion bytes of data every day. Though the data from instrumentation on aircraft, vehicles, ships, autonomous systems, simulators, and, increasingly, humans themselves does not reach this scale, its volume is significant and increasing. It is natural to want to use this wealth of data to build realistic training scenarios.

There is genuine value in using real-world data to construct training scenarios. Most obviously, it provides a degree of realism that is otherwise hard to achieve. Building the scenario entities automatically conveys another benefit as well—ease of preparation and execution. In some of the scenarios we describe below, thousands of entities are involved, and this is far more than would be feasible to create by hand. Further, depending on how thoroughly they are created manually, instructional personnel may need to monitor their behavior during the training session, which takes away from the time they could use actually instructing the students.

In this paper, we discuss our experiences building several systems that automatically build scenario entities from real-world data. One such system creates virtual scenarios that reproduce aviation mishaps, another focuses on creating realistic constructive patterns of life from real-world data, and a third concerns exploiting data from simulator-based air-to-air combat scenarios to create intelligent constructive adversaries. We conclude the paper by discussing lessons learned and general principles that will be useful to help other training professionals create realistic data-driven training scenarios.

PAST EXAMPLES

As reported in Stacy et al. (2010), we developed a prototype that sped the process of creating scenarios from flight logs with special attention to recreating aviation mishaps. The approach used a four-step process: 1) the original flight path was provided from flight data recorder data, either from the actual aircraft or from a re-creation of the mishap in a simulator; 2) on the flight path, the scenario author identified key events and pilot decision points; 3) the scenario author *generalized* the events and decision points, that is, they entered the values of key variables at those points that could also lead to the mishap. The generalization step was an acknowledgement that a safety incident will never happen exactly the same way twice—the location, altitude, speed, or other aspects of the situation may be different from the original mishap; and 4) the author connected those regions into a set of continuous envelopes describing the mishap-causing values of the key variables. These envelopes constituted the scenario. As long as pilots stayed within this envelope during the simulator mission, they encounter the circumstances involved in the mishap, and they arrived at the mishap's critical decision points. When running the scenario, instructors could choose to have

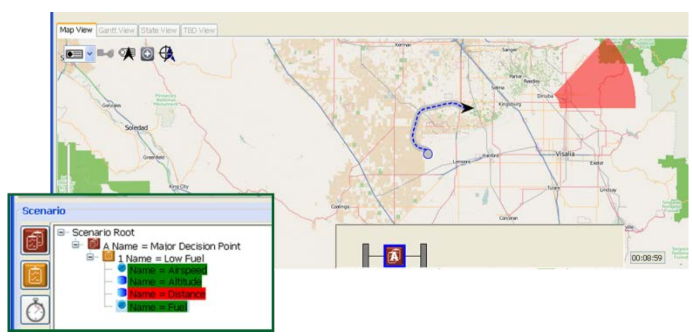


Figure 1. The instructor is notified that the trainee is outside the mishap envelope. The student is in danger of missing scenario conditions that ensure key events and decision points related to the mishap. Starting values for the mishap envelope come from real-world or simulator data, and are generalized by the instructor. (From Stacy et al., 2010.)

students encounter the mishap, or they could choose to allow the students to make decisions that would let them avoid the mishap. Figure 1 shows a screenshot from the prototype.

A second relevant example involves generating background, or "clutter," patterns of life that are qualitatively similar to real-life narrative. Clutter activities that appear to follow a narrative can increase learner sense of presence in all types of scenarios, and can be the focus of training in some scenarios. In particular, cognitive-perceptual training tasks such as observation of a crowd from a distance require narrative clutter activities (Schatz et al., 2012; Wray et al., 2015). In these efforts, narrative activities included simulated entities moving as individuals and groups within the crowd, leaving and entering buildings, and paying attention to things on their route or stopping along the way. If these activities are assembled in a random way, there is no message for the observer trainee to find. But when the activities are generated according to a narrative, the observer can learn and practice how to infer meaning from the activities in the clutter crowd, create a mental baseline of normal activity and observe changes from it, and detect deviations from the trend which indicate individuals of interest for further surveillance.

A current effort involving all the authors seeks to develop patterns of life from real-world data for use in training scenarios. The motivation for doing this is to provide a low-effort way to deploy high-fidelity patterns in training scenarios, thus freeing up training personnel to focus on instruction itself. We are using real-world data about ships to create maritime patterns of life, though we expect to use a variety of other types of data and to expand the analysis to land-based patterns of life.

What follows is a description of the processing pipeline that we have set up to automatically create these patterns of life. After discussing the pipeline, we will conclude the paper with a discussion of areas for future research, lessons learned, and the training value of easily created, high-fidelity patterns of life.

CREATING DATA-DRIVEN PATTERNS OF LIFE IN TRAINING SCENARIOS

The pipeline can be seen in Figure 2. We perform activity learning on the raw data using a variety of machine learning techniques and package the result into a language called Activity Description Language (ADL). The ADL is used as the basis for generating entities and their behavior, and the result is fed to the simulation environment for use in training sessions.

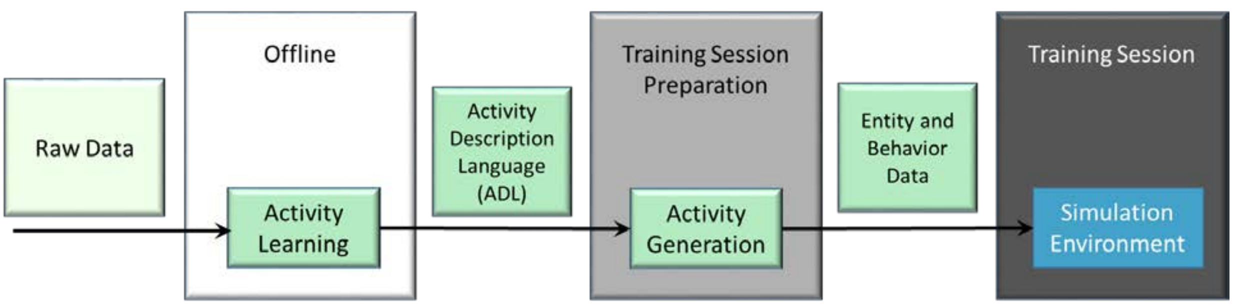


Figure 2. The pattern of life processing pipeline. Activities are learned from raw data, translated into Activity Description Language (ADL). In preparation for a training session, meaningful and realistic activities are then generated from the ADL and sent to the simulation environment. The entities and behaviors generated this way are responsive to other entities in the simulation, unlike what a simple replay of the raw data would provide.

In the next several sections, we describe the components of the pipeline in greater detail.

THE RAW DATA

Construction of patterns of life for maritime ship behavior starts with a dataset of ship behavior. This dataset should have a variety of different kinds of ships and represent a variety of different locations. The assumption was that ship behavior varied by location and ship type so, for example, ship navigation behavior in the port at Singapore is different from ship navigation behavior in the port at Seattle, which is different from ship behavior through the Strait of Hormuz. Ship behavior data over a period of time and ideally through different weather conditions and times of year would also be beneficial to represent any differences in behavior based on either time of year or weather conditions.

Ships at sea have a risk of collision and require traffic management in busy ports. To support this, a system called AIS (Automatic Identification System) has been implemented and deployed starting in the 1990s, first using short-range radio and then later using satellite transmissions. Since 2002, most ships over 300 gross tons on international voyages must have AIS transceivers. This information is not only valuable to the ships at sea, but also for other purposes and several commercial companies have emerged that provide various subsets or aggregations of collected AIS data. Table 1 shows some sample ship information from the AIS messages. We obtained AIS data for the entire globe for the month of April 2014.

Table 1. AIS Ship and Location Data Examples

Ship Information	Ship Location Information
IMO - Unique id for a ship hull	MMSI - Maritime Mobile Service identity
International radio call sign	Navigation Status such as “at anchor”, “underway using engine”
Type of ship/cargo	Speed over ground
Draught of ship	Current Position
Destination of ship	True heading
Expected time of arrival at destination	Time

AIS data provides a variety of information about individual ships and their activity. Ships typically transmit current status information such as location and speed every few seconds and transmit ship identification information every 6 minutes. Figure 3 shows a sample of the message locations by ship in two different map locations.

The data set is large and consists of almost 100 million records of ship information and over 180 million records of ship location information. Plotting the ship location data on a map provides a visual sense of the ship behaviors in different regions.

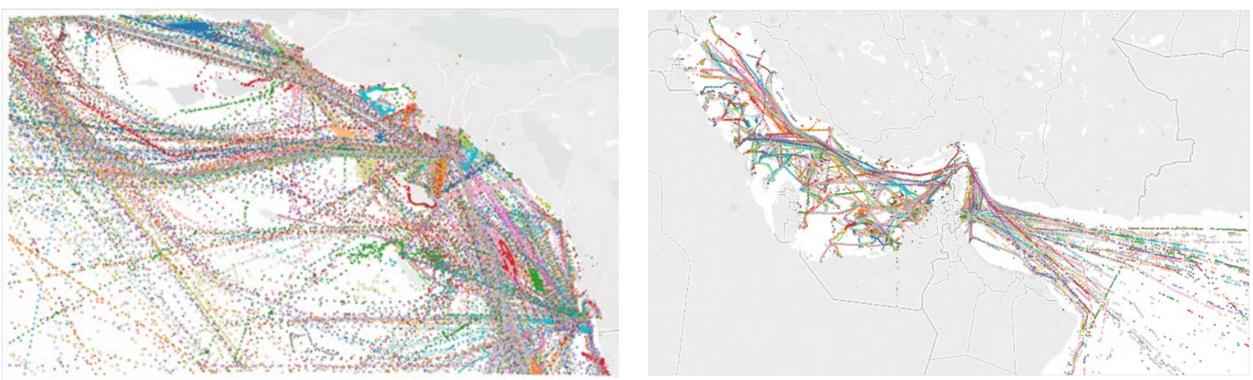


Figure 3. Map of AIS location messages plotted by ship in the southern California and Strait of Hormuz regions from April 2014. Each dot represents an AIS message, and different colors represent ships of different types. Viewed at this distance, the dots appear to form into tracks. Forming the messages into suitable tracks like these, taking care that they did not appear to cross land, was the first step in activity learning.

Table 2. Sample Ship Types and Navigation Statuses

Entity Type	Activity Type	Count	Entity Type	Activity Type	Count
Cargo	At Anchor	356	Tanker	At Anchor	427
Cargo	Moored	284	Tanker	Moored	185
Cargo	Not Under Command	57	Tanker	Not Under Command	90
Cargo	Restricted Manoeuverability	55	Tanker	Underway Sailing	34
Cargo	Underway Sailing	119	Tanker	Underway Using Engine	390
Cargo	Underway Using Engine	442	Towing	At Anchor	12
DivingOps	Moored	16	Towing	Restricted Manoeuverability	55
Dredging Or Underwater Ops	Restricted Manoeuverability	22	Towing	Underway Sailing	26
Fishing	Engaged In Fishing	28	Towing	Underway Using Engine	55
Large Towing	Restricted Manoeuverability	12	Tug	At Anchor	159
Passenger	Moored	19	Tug	Moored	243
Passenger	Not Under Command	22	Tug	Restricted Manoeuverability	206
Pilot Vessel	Underway Sailing	28	Tug	Underway Sailing	116
Pleasure Craft	Underway Sailing	15	Tug	Underway Using Engine	277

To manage this complexity, 10 regions, such as Southern California and the Persian Gulf, were identified and processed seaparately. Table 2 shows a sample of the kinds of ship types and navigation statuses that were available in one of the regions.

ACTIVITY LEARNING

To create a generalized model of ship movement, built on our previous activity learning efforts (Levchuk, Lea, & Pattipati, 2008; Levchuk, Shabarekh, & Furjanic, 2011; Levchuk, Jacobson, & Furjanic, 2013; Levchuk & Shabarekh, 2013), we took a two-phased approach. First, we compute a set of areas in the ocean within which the activities most commonly occurred. We then compute the most probable connections among the areas, for each ship activity type, where activity type is defined by a pairwise combination of ship type and navigation status. This result is a *topology* that effectively describes generalized sea lanes within which movement can occur for each activity type, as the ships navigate from their origin to their destination. For each activity type, the topologies are supplemented by activity information on probable origins and destinations, and by parameters that describe the distributions of ship density per day and ship velocities.

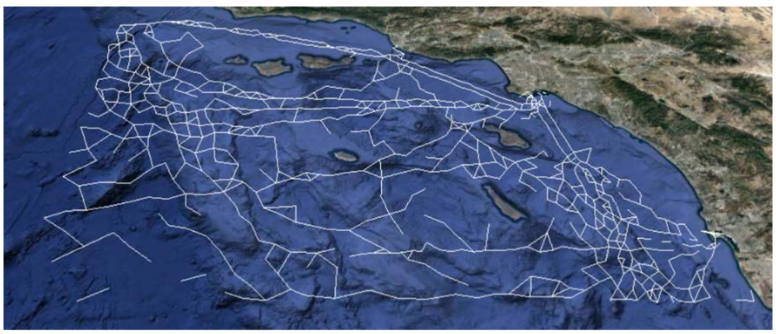
Computing the areas. Areas are described by a set of adjacent polygons that cover regions in the ocean where there was AIS activity. When traffic is heavy the areas are smaller, and when it is lighter the areas are larger. Figure 4 shows an example of the resulting areas in the ocean off the coast of Southern California.

Figure 4. Areas computed for Southern California. Areas represent concentrations of tracks formed from AIS messages, discovered using clustering algorithms. Higher density traffic resulted in smaller areas, and lower density traffic resulted in larger areas. The areas that are shown cover all activity types, but not all areas were involved in every activity type.

AIS messages contain a fair amount of variability in their usage, and this is especially true for the frequency of broadcast. We obtain ship tracks by connecting sequential AIS messages for specific ships. Unfortunately, ships that broadcast infrequently as they go around land features like peninsulas can make naively-connected tracks to appear to travel over land. For this reason, we set time and distance thresholds for connecting them—successive messages that exceed either threshold are not connected. The result is a set of tracks that are spatially and temporally connected and that always are over water.

We then create areas by clustering the waypoints in these tracks based on traffic density. For our initial set of areas, we use k-means clustering (MacQueen, 1967). The result is a set of polygons, as in Figure 4: each is a two-dimensional representation of points based on latitude and longitude. K-means clusters points relative to a center point (centroid) expressly to minimize within-cluster variance. To increase separation between clusters, we add some noise using a technique called regularization. Work in progress uses graph hashing algorithms instead of K-means clusters (cf. Liu et al., 2011), which will take advantage of spatial and temporal information in the tracks.

Computing the connections among areas. The next step is to compute the connections among the areas for each activity type. The rationale for separating connections by activity type is that different kinds of ships, with different navigation statuses, might well use different paths in the ocean. For example, cargo ships might stick more to deeper sea lanes while fishing ships might take less-traveled paths on the way to their fishing grounds, and pleasure craft might stick closer to shore. Navigation status plays a role, as well: a ship that is underway using engine might travel in different places than one that is underway sailing, or restricted by her draft; and ships that are at anchor or moored will not often be in the same locations as those that are travelling.



(a) Cargo ships underway using engine

(b) Fishing ships engaged in fishing

Figure 5. Connections among areas in Southern California for two activity types. For ease of presentation, the connections are between area centers, but they in reality the connections denote movement from anywhere in the originating area to anywhere in the destination area. Connections are computed per activity type, and, as is evident, can look markedly different for different activity types. Areas and connections together represent an activity type's topology, which is effectively a set of generalized sea lanes along which ships may travel.

To compute these connections, the tracks used for computing the areas are segmented into *tracklets* by identifying contiguous portions of the tracks that have the same navigation status. We use the tracklets to create a network of connected areas for each activity type. Figure 5 shows the connections for two ship types, (a) cargo ships underway using engine and (b) fishing ships engaged in fishing. The connections are depicted by connecting the centers of the areas, and the areas are omitted from the figures for clarity. In reality, the connections allow ships to travel from any place in an area to anyplace in the connected area, not just from polygon center to polygon center. This allows for realistic variation in ship behavior.

The areas and connections comprise a topology for each activity type, and topologies in effect define the generalized sea lanes that each individual ship can use to get from origin to destination.

BEHAVIOR ENVELOPES AND ACTIVITY DESCRIPTION LANGUAGE

Over a series of scenario-based training efforts, we have developed a formal description of learned activities (Stacy & Freeman, 2016; Jones, et al., 2015a), and it has been expressed in a language called Activity Description Language (ADL.) ADL is a language defined by an XML Schema, which means it is readable by machines and interpretable by humans. We use ADL to express the activity networks that results from activity learning, and to express additional information about those networks, including activity type densities and parameters to describe the distribution of ship velocities per activity type.

The major elements of ADL are shown in Figure 6. 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.

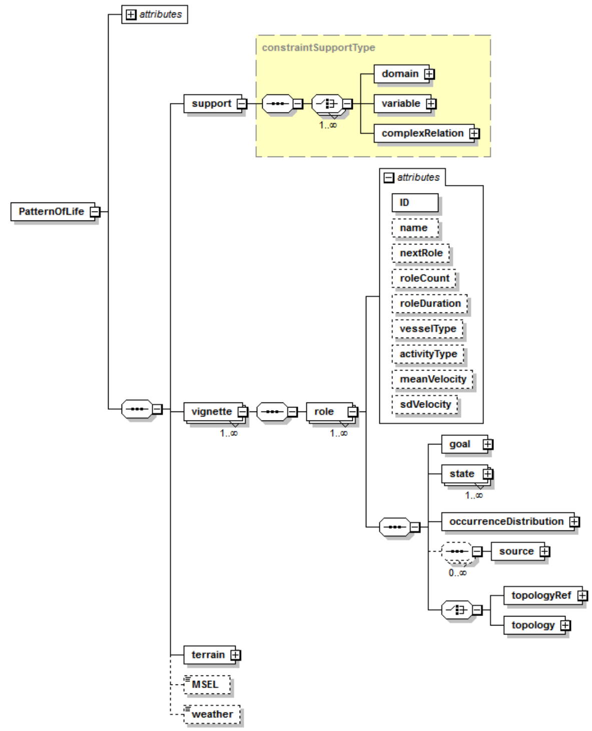


Figure 6. Top-level elements of ADL. The support element describes the variables and relations involved in constraints, which in turn describe the behavior envelopes that comprise states in a vignette. A role in a vignette has a set of goals and is effectively an activity type in the current analysis, and, among other things, it describes the activity type’s topology. Areas are described under the terrain element.

ACTIVITY GENERATION

After the behavior envelopes in ADL have been generated, they can be used to generate activities that are displayed in training simulation environments. This section describes the process of doing so.

While the high-level approach to activity generation is general, implementation details do have an impact on design. The atomic actions available in a particular simulation environment will change what activities may be generated and how they should be carried out. In order to minimize this impact, a hierarchical planner (Folsom-Kovarik et al., 2015) is used to abstract detail where possible. However, our experience suggests that the abstraction will not remove the need to know about the underlying simulation, because of different affordances and capabilities.

The present example discusses movement of ships on a geodetic surface of water. The available atomic actions in this synthetic environment allow for movement of one simulation entity from its current position to a new position in a straight line at a given speed. All other activities, such as movement along complex paths or coordinated movement of groups, must be assembled from these atomic actions.

It is easy to imagine adding actions as needed in different training domains for changing altitude, turning a body or head to a certain angle, loading people into vehicles, controlling broadcast emissions, and more actions that could exist in a simulation environment to meet specific training needs. In addition, the authors have separately used similar activity generation to support training in much more abstract domains, such as simulated cyberattacks (Nicholson et al., 2016.). Even abstract training domains still need to present trainees with an appearance of narrative and purposeful intent.

Within the example of ship movement, activity generation consists of creating ships and planning routes for them that present a coherent pattern of life. The behavior envelopes define how important aspects of these patterns may vary, because the activity generated must always stay within the relevant envelopes. Simple examples include an envelope that defines how many ships of a particular type should start the scenario within a given geographic region, what speeds are typical in this area, and what areas ships may or may not move into. Then activity generation defines where individual ships will actually move, staying within the given constraints.

A high-level view of our activity generation sub-pipeline is presented in Figure 7 below. Activity planning takes high-level goals specified in behavior envelopes and produces a sequence of specific actions that occur in an entity’s lifecycle. Next, this plan is refined at increasingly high levels of fidelity until it meets all local and/or environmental constraints (for example, ensuring generated planes follow appropriate altitude blocks in a tactical air domain, or generated ship trajectories always lie on surface water). Finally, these low-level actions are translated into atomic actions specific to the target simulation. This hierarchical refinement process allows even complex behaviors to be embedded in training simulation environments that have only basic entity controls available.

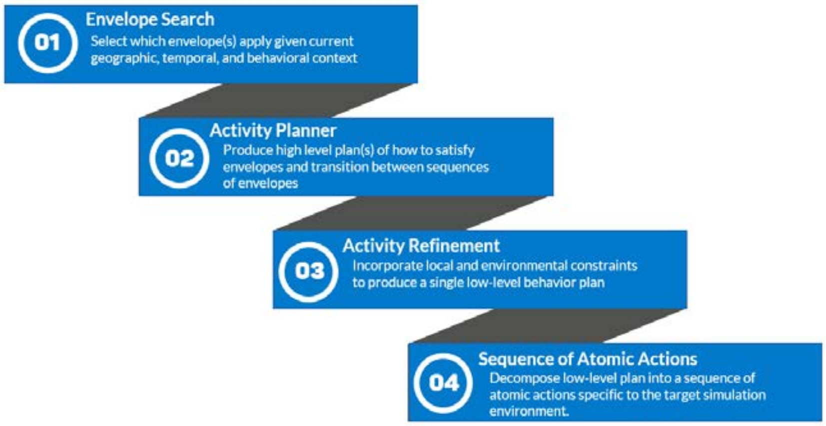


Figure 7. The activity generation pipeline. The applicable behavior envelopes are identified, and planning is performed at a high level. Because the activities learned from AIS may not have had the required granularity to show local behaviors, these activities are refined into a smooth behavior plan, which is then translated to a set of simulation-environment-specific atomic actions.

A key contribution of activity generation is providing underlying intent to simulated entities. That is, each entity will not simply move randomly within the given boundaries. Random movement does not provide the interpretable cues that trainees need to be able to perceive for training purposes. Instead, individual movements are planned in the hierarchical planner so that they sum up to subgoals and an overarching goal of the entity. For example, one fishing ship might have a goal to leave home, fish for a day, and return with the catch. Although the goal is never communicated directly, trainees can infer it from the individual movements that are generated in the simulation. For example, the fishing ship starts moving in the early morning and not late in the day, it moves quickly to a fishing area and then moves slowly as it trawls, and when the ship returns home it moves to the same place it came from not a different dock location. Tractable planning of the hierarchical goals, subgoals, and atomic actions for a large population is a computational challenge we have previously discussed elsewhere (Jones et al., 2015b).

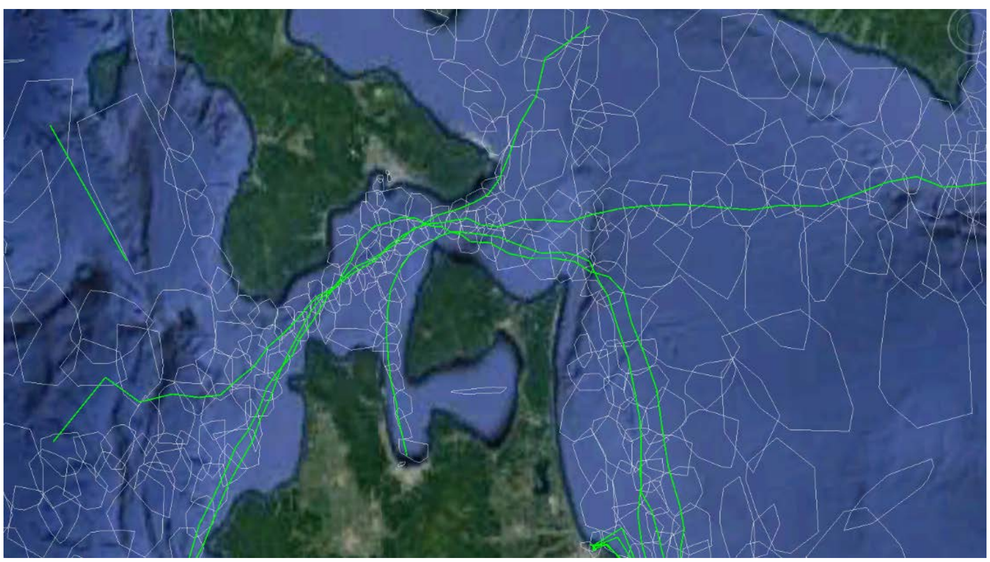


Figure 8. The resulting activities. The output of activity generation is shown by the green, within the areas from the behavior envelopes that describe activity type-specific movement, shown in white.

To carry out its goals within an array of behavior envelopes, our activity generation uses two fundamental methods to combine multiple behavior envelopes: progression and composition. Progression refers to planning how to move from one active envelope or set of envelopes to another. It is accomplished by the hierarchical planner. Composition refers to composing or combining multiple envelopes that are relevant to the same simulation entity. For example, two separate behavior envelopes might specify (1) top speed of a cargo ship is 20 knots and (2) the speed within this geographical region is limited to produce no wake. An entity might find that one, the other, or both envelopes govern its movement speed at any given time. In real training, many envelopes at once might partially overlap in this way. To address this, we compose input envelopes during activity generation and find a partition of the activity space into smaller, artificially generated sub-envelopes which do not overlap. From the point of view of any particular cargo ship, it is either in a sub-envelope where its top speed is 20 knots, or it is in a sub-envelope with a limited speed, but never both. By applying envelope composition to the many envelopes that govern an entire training simulation, it becomes tractable to apply progression through the sub-envelopes and plan the movement of each entity.

We close with an illustration of the realism inherent in the result of activity generation. Figure 8 depicts a coastal area with ports and a visualization of the behavior envelopes that were learned from real-world data for that area. The areas in the behavior envelopes, depicted as white polygons, describe areas where ships are likely to be found and connections in the behavior envelopes that descript how to move between areas (not shown). Based on these inputs, the activity generation sub-pipeline created cargo ships that move along the green paths in the figure. These paths can be seen to move in the same areas as the real ships did, move in and out of ports on the map, and move with inferable intention.

THE TRAINING VALUE OF PATTERNS OF LIFE FROM DATA

Our validation efforts are ongoing, but it is worthwhile to report some preliminary results here. The goals of the validation are to establish both 1) *an increase in the fidelity* of white shipping patterns of life and 2) *a decrease in the effort* that instructional personnel need to spend to create and execute them in a training environment, and to accomplish both of these things with an improvement to training effectiveness.

Certain training objectives (TOs) from certain domains require high-fidelity background shipping activity. This is true, for example, for certain TOs in Intelligence, Reconnaissance, and Surveillance missions that require detecting anomalous behavior, and for other TOs in Anti-Submarine Warfare that require realistic acoustic environments. The current approach to preparing scenarios for these TOs is labor-intensive: scenario designers visit a relevant AIS web sites for reference and work up descriptions of 100-150 commercial ships in the area of interest. Executing the scenario is labor-intensive as well, because instructional personnel must monitor those ships to ensure they show plausible behavior. This is time taken away from actual instruction during the session. Further, in some cases a realistic number of ships might be 1000 or more, which is an infeasible number to create or execute manually.

On the other hand, technologies exist to quickly generate large numbers of ships in a scenario—it is just that the resulting ship behavior is not very realistic, but rather is controlled by a handful of simple rules. For some TOs, those that do not depend on high-fidelity background traffic, this is fine; but when it is important to be able to identify anomalous behavior, or to learn to work in realistic acoustic environments, more realistic white shipping is a requirement.

Preliminary reports from instructors are enthusiastic about the prospect of being able to create realistic maritime traffic easily and automatically, primarily because they look forward to being able to spend their time actually instructing the trainees rather than maintaining the scenario. The general strategy of providing realistic patterns of life as intelligent background in training scenarios, of course, is not limited to the maritime arena; a similar strategy can be used for any training arena, whenever TOs require accuracy in background activity. We believe that this capability will lead to improved training effectiveness and efficiency, and ultimately to a higher level of readiness.

LESSONS LEARNED

- **Be prepared for big variations in coverage and quality of the data sources.** Creating realistic patterns of life from data very much depends on the availability and quality of data. In a sense, maritime patterns of life are relatively easy because of the widespread use and availability of AIS data, though real-world AIS data has a surprising amount of noise and other unexpected "features." Not only was there large variability in the frequency of message broadcast, but there were also free-text fields that users filled in surprising ways. For example, one ship's destination field read, "**ARMED ESCORT**", presumably to discourage piracy.

  In addition, the granularity of AIS data was not always adequate to infer smaller-scale behavior like docking in a port with the assistance of tugboats. Fortunately, the activity generation portion of our pipeline is capable of filling in the blanks in a realistic manner for such behaviors.

  For traffic data, there is generally good coverage in some areas of the world like the U.S. and Europe, but other regions of the world either have no traffic data or do not want to make it available outside the country. Pedestrian data is very sparse, generally limited to a specific neighborhood in a U.S. city. This means that for most land-based patterns of life it will be necessary to do some additional modeling to provide approximations for areas in which there is no coverage.
- **The very real virtues of activity learning aren't always apparent to casual observers.** Initially, when learning about the activity learning-activity generation pipeline, casual observers sometimes ask whether a simple replay would suffice. For most application, the answer is "no." First, the replay of the pattern of life will not change no

matter what else happens in the scenario. The pattern of life is limited to exactly what happened in that particular place at that particular time. Entities cannot react to or interact with other "live" entities around them. Second, the fact that there may be noise and inconsistencies in the data, and that there will be holes in coverage, means that entities may appear to behave oddly or may disappear once certain geographic boundaries are crossed. Finally, there will be no opportunity to display variability in the pattern of life—it will always be exactly the same for every use of the scenario. Variability is in general valuable for training (cf. Schmidt & Bjork, 1992), especially for retention and transfer to new tasks, and in any case, experienced students may be able to game the system since they will be able to know exactly how the pattern of life will behave. For these reasons, activity learning and generation are an integral part of providing an effective training environment when generating scenarios from data.

- **Scalability matters.** The AIS data involved nearly 400,000 ships worldwide. While it would be an unusual scenario that would cover the entire globe or that would need all of the ships to participate in a pattern of life, the fact is that a large number of ships were involved in each of the ten regions we analyzed. In some cases, it was necessary to tune our activity learning algorithms in order to fully analyze a region; and generating activities for thousands of ships required taking a special approach to specifying behavior in the simulation environment so as not to overwhelm the systems it was running on. In our experience, it is definitely worth understanding and adapting to the scalability constraints of the activity learning and generation environments.
- **The human visual system easily spots patterns that are challenging algorithmically.** It is often interesting to view the data at a low enough granularity that patterns emerge for individual ships. For example, we found a ship track that crossed a region of the ocean in a back-and-forth zigzag manner. On further investigation, we discovered that it was a research ship, likely engaging in an activity like search or making precise bathymetry measurements. Patterns like these are easy for people to spot, because they stand out from other, more mundane tracks, but in general, algorithmically finding meaningful anomalous patterns in large data sets, and especially interpreting them, is an area of active research in the pattern recognition community, and it remains challenging.

CONCLUSIONS AND FUTURE RESEARCH

One obvious avenue for future research is the application of the pattern of life pipeline to other data. Our initial effort will focus on using land-based vehicle traffic data to create traffic patterns of life, but eventually we expect to be able to use data describing trains, commercial aircraft, and pedestrians. Ultimately, we expect to be able to use all the data in conjunction with narrative accounts of missions and other events to automatically provide a rich set of meaningful scenario behaviors.

Future research also includes improved real-time responsiveness of activities. When learners not only observe but interact with and disturb the generated activities, the responsiveness must be quite robust in order to support all anticipated and unanticipated inputs. While the current activity generation supports responding to changes within an expected range of input, there is an interesting research question surrounding response when the learner makes an unanticipated choice. The generated activities should appear to respond reasonably and, ideally, should act to return the scenario flow back to a channel that is anticipated and will show useful training. The envelope representation of behaviors may enable efficient definition of how to act when outside any expected envelope in one or more dimensions. With proper nesting and prioritization of such envelopes, it may be possible to make activities more robust to the kinds of disruption that the human element introduces in any training scenario.

Complex, realistic, easily constructed patterns of life are valuable for training, but for the most part, they will be in the background during the training scenario. Another research goal will be to extend the technology to be able to create foreground entities that are directly related to training objectives, as in Stacy & Freeman (2016.)

Despite the fact that we have been able to generate realistic maritime patterns of life, we believe that techniques for creating scenarios from real-world data are still in the early stages. We look forward to their continued development, and to the dramatic improvements in realism and convenience in complex scenario-based training that will result.

ACKNOWLEDGEMENTS

The views expressed herein are those of the authors and do not necessarily reflect the official position of the organizations with which they are affiliated. We would like to acknowledge the Office of Naval Research (ONR) and the Naval Air Warfare Center Training Systems Division (NAWCTSD) for their support of this work. CDR Brent Olde of ONR provided wonderful encouragement and advice, and Dr. Heather Priest of NAWCTSD worked diligently to provide collegial guidance and collaboration. This study received strong operational support from John DiCola and his team at the Air Combat Environment Test and Evaluation Facility at Patuxent Naval Air Station, as well as Sam

“Scrammy” Parmarter from BGI, Inc., whose excellent support has ensured that the patterns of life we develop are compatible with the Next Generation Threat System. We would also like to acknowledge strong support from Mark Bunn and his team from ASEC, Inc., Alyssa Mercado from NAWCTSD, and Kenny Riddle from Aptima, Inc. whose work on validating the system has been invaluable. Finally, Ron Smits from Engility provided helpful comments on earlier drafts of this paper.

REFERENCES

Folsom-Kovarik, J.T., Woods, A., Jones, R.M., & Wray, R.E. (2015). Narrative representation for training cross-cultural interaction: Lessons learned and recommendations. *Proceedings of the 6th International Conference on Applied Human Factors and Ergonomics*, Las Vegas, NV.

Jones, R. M., Bachelor, B., Stacy, W., Colonna-Romano, J., & Wray, R. E. (2015a). Automated monitoring and validation of synthetic intelligent behavior. In *Proceedings of the 17th International Conference on Artificial Intelligence*. Las Vegas.

Jones, R.M., Folsom-Kovarik, J.T., McLaughlin, P., & Frederiksen, R. (2015b). High-density pattern-of-life modeling. *Proceedings of the International Conference on Artificial Intelligence*, Las Vegas, NV.

Levchuk, G.M., Jacobson, M., & Furjanic, C. (2013). Learning and Detecting Coordinated Multi-Entity Activities from Persistent Surveillance. *Proceedings of the SPIE Defense, Security, and Sensing Conference.*

Levchuk, G.M., Lea, D., & Pattipati, K.M. (2008). Recognition of Coordinated Adversarial Behaviors from Multi-Source Information. *Proceedings of the SPIE Defense, Security, and Sensing Conference*.

Levchuk., G.M., & Shabarekh, C. (2013). Fusing video and text data by integrating appearance and behavior similarity. *Proceedings of the SPIE Defense, Security, and Sensing Conference.*

Levchuk, G., Shabarekh, C., & Furjanic, C. (2011). Wide-threat detection: Recognition of adversarial missions and activity patterns in Empire Challenge 2009. *Proceedings of the SPIE Defense, Security, and Sensing Conference*.

Liu, W., Wang, J., Kumar, S., & Chang, S.-F. (2011). Hashing with graphs. In *Proceedings of the 28th International Conference on Machine Learning*. Bellevue, WA.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, pp. 281–297.

Nicholson, D., Massey, L., O'Grady, R., & Ortiz, E. (2016). Tailored cybersecurity training in LVC environments. *MODSIM World Conference, Virginia Beach, VA.*

Schatz, S., Wray, R.E., Folsom-Kovarik, J.T., & Nicholson, D. (2012). Adaptive perceptual training in a virtual environment. *Proceedings of the 56th Human Factors and Ergonomics Society Annual Meeting*, Boston, MA.

Schmidt, R. A., & Bjork, R. A. (1992). New conceptualizations of practice: common principles in three paradigms suggest new concepts for training. *Psychological Science*, **3**, 207-217.

Stacy, W. & Freeman, J. (2016). Training objective packages: enhancing the effectiveness of experiential training. *Theoretical Issues in Ergonomics Science*, **17**(2), 149-168.

Stacy, W., Picciano, P., Sullivan, K., & Sidman, J. (2010). From flight logs to scenarios: flying simulated mishaps. *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference*, Orlando, FL.

Wray, R.E., Folsom-Kovarik, J.T., Woods, A., & Jones, R.M. (2015). Motivating narrative representation for training cross-cultural interaction. *Proceedings of the 4th International Conference on Cross-Cultural Decision Making*, Las Vegas, NV.