

Building a Game to Educate Senior Officers in Counter-Piracy

Perry McDowell, R. Erik Johnson
Delta3D, The MOVES Institute
Monterey, California
<mcdowell/rejohnso>@nps.edu

Jared Freeman, Matthew Roberts, Zachary Horn
Aptima, Inc.
Woburn, Massachusetts
<freeman/mroberts/zhorn>@aptima.com

ABSTRACT

The MOVES Institute and Aptima, Inc., teamed to create a game for use at the Navy's Surface Warfare Officers School (SWOS). SWOS will use the game in the Prospective Commanding Officers (PCOs) class to teach as well as evaluate students in counter-piracy procedures.

Building an effective game for learning in the counter-piracy domain provided many challenges. SWOS wanted the game to assess students in ambiguous areas where there was not necessarily one correct answer. The game also needed to stress the students by putting them into positions where the book did not provide definitive guidance. Additionally, the game needed to evaluate the students' performance in the game, since SWOS does not have enough instructors to observe the students playing the game. Finally, the game had to be easy to modify after delivery so that SWOS could update the game as changes occurred in tactics and procedure.

This paper describes the procedure used to create the game, the game itself, the performance measure system used to evaluate the students and provide feedback, and the tools created to allow easy modification of the game in the future.

ABOUT THE AUTHORS

Perry McDowell is a Research Associate at MOVES Institute at the Naval Postgraduate School in Monterey and the Executive Director for the Delta3D Game Engine Project. He has been on the faculty of the Naval Postgraduate School since 2000, where he teaches classes in Computer Graphics and Virtual Environments.

He has primarily focused his research on game based training and open source technologies. Currently, he is researching the best ways to implement game based learning into various military curricula.

Mr. McDowell is a former Naval Nuclear Power – Surface Warfare Officer, whose sea tours include USS-VIRGINIA (CGN-38), USS ELROD (FFG-55), and USS ENTERPRISE (CVN-65). He graduated with a B.S. in Naval Architecture from the U.S. Naval Academy in 1988, and an M.S. (with honors) from the Naval Postgraduate School in 1995.

Jared Freeman is the Chief Research Officer at Aptima, Inc. He has extensive experience investigating problem solving by individuals and teams in real-world settings. Specifically, Dr. Freeman develops decision aids, training content and systems, performance and communications measures, feedback technologies, and organizational structures that support mission leaders and their staffs. Dr. Freeman is the author of more than 90 articles and book chapters concerning these and related topics.

Dr. Freeman holds a Ph.D. in Human Learning and Cognition from Columbia University and a M.A. in Educational Technology from Teachers College, Columbia University.

Matthew Roberts is a Principal Software Engineer and Team Lead of the Performance Assessment Systems team at Aptima, Inc. Mr. Roberts has experience leading teams of software engineers as they support a variety of concurrent projects focused on developing performance measurement and assessment systems and computer based training systems. Mr. Roberts has extensive experience architecting and developing automated system based measurement solutions and has lead the integration of Aptima's measurement technologies with multiple DoD distributed

simulation environments. His prior software development experience is broad, and ranges from designing database driven web services, to creating report authoring tools for medical information systems, to developing file system filter drivers and multi process anti-hack software.

Mr. Roberts holds a B.S. in Computer and Systems Engineering from Rensselaer Polytechnic Institute.

Erik Johnson is a Research Associate at the Modeling, Virtual Environments, and Simulations Institute (MOVES). Mr. Johnson graduated from the Embry-Riddle Aeronautical University in 1995 with a Bachelor of Science in Aviation Computer Science. After graduating, he was a key software engineer for Boeing Helicopters in Mesa, Arizona where he helped design and develop real-time graphical simulations for future rotorcraft designs. In 2001, he took the position of Technical Director for the Human Performance Engineering and Game-based Simulation group at the MOVES Institute. Mr. Johnson is the founder of the Delta3D Open Source Game and Simulation Engine and is actively managing the engineering efforts.

Zachary Horn is an Industrial-Organizational Psychologist at Aptima, Inc. His work includes designing workshop and web-based training tools, developing computer-based training aides for critical decision making, training validation, and advancing the assessment and measurement of on-the-job learning. Dr. Horn received a Ph.D. and M.A. in Industrial-Organizational Psychology from George Mason University, and a B.A. in Psychology from the University of California, Santa Cruz.

Building a Game to Educate Senior Officers in Counter-Piracy

Perry McDowell, R. Erik Johnson
Delta3D, The MOVES Institute
Monterey, California
<mcdowell/rejohnso>@nps.edu

Jared Freeman, Matt Roberts, Zachary Horn
Aptima, Inc.
Woburn, Massachusetts
<freeman/mroberts/zhorn>@aptima.com

Introduction

In the past few years, the Navy has returned to its roots by engaging once again a familiar scourge: piracy. At the end of the eighteenth century, Congress approved the building of six frigates to protect American shipping in the Mediterranean Sea against African pirates (Beach, 1986). Today, the Navy, along with its coalition partners, is again waging a vigorous counter-piracy (CP) campaign, this time primarily off the east coast of Africa.

Although the mission is one of the Navy's oldest, the irregular nature and rapid evolution of CP make any domain-related training and assessment a difficult challenge. CP is very similar to other aspects of maritime interdiction operations (MIO), such as counter-narcotics, or enforcing sanctions under the United Nations resolutions against the Saddam Hussein regime, that the Navy has performed in the last decades. However, CP has some unique differences that developing Navy officers must learn before they reach their ships, as there is an increasing likelihood that their ship will be involved in CP during their next sea tour.

The primary location for training surface warfare officers (SWOs) before they reach their ships is Surface Warfare Officers School (SWOS) in Newport, Rhode Island. SWOS recognized the need for a method to train senior SWOs – executive officers and commanding officers of ships – in the intricacies of CP, and requested new training from Office of Naval Research (ONR) to assist this effort. The TechSolutions branch of ONR chose the Delta3D team, part of the MOVES Institute at the Naval Postgraduate School, and Aptima, Inc. to build a game-based training system to develop decision-making skills in CP for prospective commanding officers (PCOs) of warships. This paper covers the development of *Back to Basics* (B2B), a game to give PCOs a no-risk simulated environment in which to experience the decision-making they will need to exercise in the real world shortly after training at SWOS. In B2B, instructors can remediate errors made by students that

would otherwise have drastic real-world consequences in terms of lives, personnel, and money.

Examination of Domain and Learning Objectives

Personnel from both Aptima and Delta3D investigated the ill-defined domain of CP to identify learning objectives on which to evaluate PCO decision making. Examinations of current course materials and learning objectives assisted greatly in developing a functional knowledge of the domain. An example of a current course-based learning objective is to “describe available MIO assets and platforms, their employment considerations, MIO tactics, MIO intelligence requirements, MIO communications considerations, and [Rules of Engagement]/Use of Force considerations” (Surface Warfare Officer School, 2010). The declarative nature of the course-based learning objectives provided several standard criteria that helped inform the contextual design of the B2B game. However, this system required a new set of learning objectives to assess the *application* of this knowledge to the prevention and cessation of piracy in a simulation.

Through discussions with SWOS instructors, it became clear that given the complexity and ambiguous nature of CP events, there is seldom a single correct response. In these situations, the Commanding Officer must exercise judgment based often on limited contextual information, thus creating an environment in which success is contingent upon the decision maker's ability to identify risks and quickly establish a creative response to the problem at hand. Thus, the game needs to present situations that reflect reality; situations in which there might not be only *one* “correct” answer, but in which a myriad of possible solutions exist to achieve mission goals successfully. As one SWOS instructor noted:

“The game should not test the PPRs (pre-planned responses) – PCOs know those... It needs to present the situations where our students will be earning their command pay” (LCDR R. Portillo, personal communication, June 2, 2010).

The system must address the types of complex CP problems that course-based training does not cover. For example, a boarding team is aboard a suspect vessel and the tension is increasing. The team has control of the situation, but the helo providing valuable assistance must land to refuel in 15 minutes. The ship will have to change course away from the boarded vessel to align with the prevailing winds and receive the helo for refueling. This valuable support for the boarding team will be unavailable as the tension continues to grow aboard the vessel. As commanding officer of the ship, how does the PCO respond? Other sample events requiring command judgment include:

- A breakdown of the ship's boat with the boarding team aboard the suspect vessel;
- A vessel crew member falls overboard;
- An accidental weapons discharge injures a vessel crewmember.

Multiple creative solutions exist in these situations, all of which may be technically acceptable and yet debated by subject matter experts with different experiences and perspectives. This makes the design of the game and the related performance measures especially challenging for use in officer training. The software must adopt a measurement approach that is more complex than typical "yes/no" assessments by defining performance in relation to the contextual cues that exist as the decisions are made.

The result of this investigation was a list of four primary learning objectives for applying command judgment during a CP mission:

1. *Mission Analysis*: identification and exploration of suspect vessels and possible threats;
2. *Mission Execution*: achieving the mission in accordance with rules of engagement and standard operating procedures, if applicable;
3. *Application of Force*: appropriate use of force for prevention and cessation of piracy acts;
4. *Command Leadership*: appropriate and justifiable responses to emergent events that are both novel and surprising.

Each learning objective was critical in determining criteria for building a game environment that enables SWOS instructors to easily modify/author CP scenarios, and for creating advanced performance measures on which to assess PCO performance.

Throughout the remainder of this paper, we will show how the irregular and continually evolving mission of CP became a primary driver in designing the game,

performance measures, and scenario authoring tools. Several high level game design requirements were identified in relation to this unique CP domain, SWOS needs, and the constraints imposed by SWOS resources. The six primary requirements that drove the development of B2B are:

- The game shall challenge the student with ambiguous scenarios;
- The student shall be able to play the game with minimal training;
- The game shall minimize the need for instructor oversight and the time students spend playing
- The instructor shall be able to create new scenarios using a simple interface;
- SWOS personnel shall be able to modify the game to reflect changes in SOPs;
- The game shall run on the technological infrastructure in place at SWOS.

Once the examination of the domain and the learning objectives was completed, the teams began working in their separate areas, keeping in close contact to ensure successful integration. We will discuss how the integration was accomplished later in this paper.

Game Development

Game Design Issues

One of our major design decisions was whether to create an "open world" game or a linear game. An open world game is one in which the player has multiple paths to complete each challenge, while in a linear game the player has only a single path for task completion. For example, in a linear game the player must overcome obstacle A before obstacle B, obstacle B prior to C, etc. In an open world game, obstacles can be completed in any order desired by the player. Examples of open world games series include the *Grand Theft Auto* and *Legend of Zelda* series, while linear games include the *Call of Duty* series and *Gears of War* series.

Open world games are more difficult to create, since the developers cannot know in advance the player's state at a given point in the game. For example, in a linear game, a player might need an amulet to defeat a non-player character in level 3; if the developer requires the player to get the amulet before he leaves level 2, the developer is certain the player can complete the task.

The putative advantage of open worlds is that the player can do anything at any time, allowing for great

creativity in the development of an adaptive response. For example, in a famous Star Trek episode where the Romulans seriously damage the ENTERPRISE, Captain Kirk decides that the only hope for survival is to send a message containing false information using a code he knows that the Romulans will intercept. The message informs Star Fleet that he will destroy the Romulan ship from a great distance using a device that, in reality, does not exist. The Romulans decode the message and leave, allowing the ENTERPRISE to escape. (Deadly Years, n.d.) The creative use of deception by Captain Kirk was both brilliant *and* possible, albeit unforeseen. The possibility for creative command responses, which is similarly present for commanding officers in modern-day CP situations, pose difficulties for a game developer building an open world game.

In order to provide the level of freedom in decision-making that SWOS intended for PCOs in training, the decision was made to produce B2B as an open world game. While the chosen game format reflects the art of being a commanding officer, this decision made both the design of the game and evaluation of the students more difficult.

In order for Kirk's ruse to work in a game, the game must include many innovative design and coding elements. These would include the ability to send a message to Star Fleet describing a non-existent weapons system; the odds that the Romulans will intercept the message; and the ability for a Romulan ship – using artificial intelligence – to correctly interpret and respond to the consequences of the creative message. Game design was similarly difficult in B2B because we had to consider many factors that might be involved in an ill-defined counter-piracy context.

For example, the master of a suspected pirate vessel might comply with a standard boarding, but react with hostility if the PCO performs a dangerous ship maneuver close to the suspect vessel during boarding. To build the same reaction into the game, the developers would need to add to the game the knowledge that the master's attitude would change after such a maneuver, the ability to differentiate between safe and dangerous ship maneuvers, and create the NPC's AI to react appropriately to both types of movements. It is admittedly impossible to include all possible combinations of maneuvers and events. However, close collaboration with the subject matter experts at SWOS ensured that B2B would provide PCOs with the creative freedom to construct many of the innovative solutions they might use in the real world.

Defining the required set of elements for such a game was driven largely by domain knowledge and the learning objectives specified earlier. For example, because most officers at the O-5 level are making operational level decisions and may not see the enemy, simulations to train them could possibly use a real-time strategy (RTS) game approach. RTS games are often map-based, with the player receiving contextual information in the form of event reports rather than through direct observation. Therefore, a game view set entirely in the Combat Information Center (CIC) of a ship was considered, in which the student observes radar scopes and receives messages from members of the crew. However, that solution would not have had the desired training effect, as many of the decisions made by a CO in CP missions require a direct line of sight to the suspect vessel.

The CO must decide the boarding team's weapons posture, whether the situation requires additional weapons coverage (e.g., launching a helo), as well as whether the suspect vessel is seaworthy enough to safely deploy his boarding team. Training requirements provided by SWOS confirmed that the CO makes these decisions primarily by his or her own direct line of sight. Decisions to board are influenced by the actions and attitudes of the suspect vessel's crew, the vessel's appearance and motion through the seas, and many other visual cues. This required the game to have a realistic 3D visualization component.

During CP events, the commanding officers do not actually "perform" specific tasks; instead, they give orders to the crew, which responds by performing those specific tasks. This required the addition of layers of indirection to the game control to simulate what would actually occur, thus complicating the chain of events involved in many tasks. Without this indirection, the student would fire at an object by approaching a .50 caliber machine gun on the deck of the ship. Instead, to mimic reality, the game was designed such that the student provides an order to fire at a specific target, which is then carried out by a member of the crew.

Similarly, a ship's CO is not responsible for determining the course required to intercept a vessel or the wind direction required to launch a helicopter. Rather, the CO provides orders to the Officer of the Deck (OOD) who ensures that all the required conditions are met. Therefore, the game also needed to realistically accomplish these tasks.

The requirement that the game had to run on existing computers in the SWOS classroom presented additional design constraints. These computers were

not equipped to handle the rigors of today's graphic-heavy simulations, as they did not contain graphics cards. This limited development options, as modern game engines like Delta3D use graphics cards to handle their most powerful features. For example, Delta3D has a robust ocean model, but the model is ineligible for use because it performs its wave calculations using the GPU. Thus, to visualize current sea state – an important factor in CP decision making for PCOs – we had to spend time creating a lower-fidelity method to convey this information to the player. Additionally, reduced GPU power limited both the visual realism of the scenes and the animations of the avatars in the world.

Finally, the game must effectively balance between moments of tedium and moments of action, as this balance is the real source of learning for commanding officers. Both to save the students' time and to maintain student interest, the game should adequately compress those hours of tedium while still allowing enough time for the students to make decisions. To do this, we allowed the student to increase the passing of game time up to a factor of sixty-four, meaning that an hour of game time could take just under a minute of actual time. One problem with advancing time quickly is that the event periods requiring a student's response can move too quickly to allow a complete response. To overcome this, whenever the game alerted the student to an event, it reset the speed ratio to 1:1, allowing the students to make decisions in real time.

Game Design

We mentioned earlier the critical need for the PCOs to see events unfold through their own eyes, and this dictated placing the students in a 3D world. However, much of the information comes into the ship via its own sensors, such as radar, or data transmissions from other Navy ships, both of which are displayed in 2D. Therefore, we decided to have a hybrid interface, which allowed the player to choose between 2D and 3D as the situation warranted.

Above both the 2D and 3D displays, the application shows the information bar (shown in Figure 1). This bar shows the information the student will need in both displays, such as course, speed, wind speed and direction. It also contains controls for location and time. The student moves to a different location or view using the location controls, and determines how quickly time passes using the time controls.

The 3D display simulates the two bridgewings of a ship, which is where a PCO would spend most of his or her time during a CP operation once the suspect vessel

is in sight. On either bridgewing, the student can use the mouse to change the view to any direction from directly ahead to directly astern. At the bottom of this view are a bearing circle, showing true bearing, and a digital display of relative bearing, both of which are important for the PCO to maneuver the ship. On the bridgewing, the student can choose to look at a magnified view, similar to looking through binoculars.

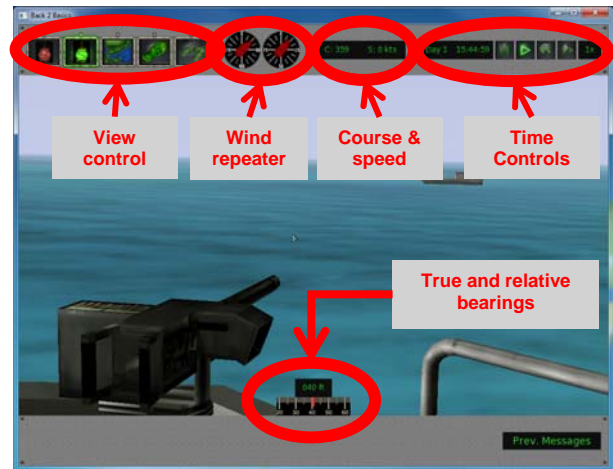


Figure 1. Bridgewing view of *Back to Basics*

Students primarily use the 2D display before the ship has visual contact with the suspect vessel. The display is similar to a *Global Command and Control System – Maritime* (GCCS-M) and uses the same symbology to display other vessels and aircraft, making it easy for the students to interpret without any training. In addition to giving the students information about the situation, it is also where the student issues orders. The student selects a symbol for a vessel and then chooses what he or she wants done to that vessel. This serves as a level of indirection, allowing the student to make clear to which vessel the actions should be applied. Figure 2 shows the 2D view, along with the menu system.

Within this framework, the game passes information to the student and the student acts by using the control system. For example, if the student receives a call indicating a pirate attack on a vessel, he or she can order the ship to intercept the vessel, inform higher authority and launch the helo. Once on the scene, the student orders the boarding team to board the suspect vessel and they report what they find. Based upon their reports, the student recommends a course of action to higher authority. If any anomalies occur, the student either gives orders to handle them or dismisses them as irrelevant.



Figure 2. 2D View of *Back to Basics*

Intelligent Tutoring System

The B2B game was designed to provide an environment in which PCOs can experience decision making during counter piracy operations, and to meet the 6 high level requirements laid out above. These requirements include both minimizing the workload for the instructors and minimizing the required classroom time for the students. To meet these requirements, capabilities were needed in B2B that allowed it to function as an intelligent tutoring system without the assistance of instructors in every training session. The included both automated performance assessment and automated methods to select the sequence of experiences that would most benefit the student based on their performance.

Effective scenario-based training requires that students receive feedback on their performance. Ideally, instructors would provide this instructional feedback. However, when the classroom environment consists of a large set of stand-alone simulators, the instructor can only reasonably track one student's progress at a time. Additionally, training curriculums typically involve each student progressing through the course on the same path and at the pace. In many cases, this strategy leads to students being over or under challenged.

The requirement to minimize instructor workload suggested the addition of automated system-based performance measurement to assess decisions made during the simulation. The assessments are used to provide feedback to the students as well as providing the basis for selecting of subsequent scenarios so that each student's time in the classroom is utilized efficiently. This is essentially an intelligent tutoring approach that assesses students as they progress and

provides them with the experiences they need for maximum learning.

To provide automated assessment, B2B employs Aptima's Performance Measurement Engine (PM Engine™) to automatically calculate performance measures and provide assessments of the PCO's decision making during the training. The PM Engine is an application that operates on simulation data and measures performance in real time. The PM Engine is configured with Human Performance Markup Language (HPML), an XML based language developed by Aptima which allows for the description of trainees, contexts, measures, assessments and other training related objects in a format that is both human and machine readable. Once the PM Engine has been configured, it listens to simulation data and assesses trainee performance in real time. These assessments are then published to a database where they are available for inclusion in a variety of reports and can be provided to the trainee as feedback during a debrief. Because objective system-based measures available to summarize the training event and the student's performance, the instructors can be relieved from having to watch the entire training event take place and can focus solely on the debrief. This is significant for a SWOS instructor who regularly has a class of over 40 students.

An example of a measure, which assesses the PCO's decision-making and more specifically their attention to the boarding crew's safety while making decisions and which can be applied to the scenario described in the *Examination of Domain and Learning Objectives*, is the *Boarding Crew Support Measure*. The *Boarding Crew Support Measure*, which continuously monitors if either a helo or ship is in a support position while a boarding team is aboard a suspect vessel, assesses the PCO's command judgment for the *Mission Execution* learning objective. It is important to note that all measures can be configured, as seen in Figure 3, during scenario configuration to reflect the expected or acceptable courses of action for the specific scenario that they are being applied to. For instance the measure can be configured so that support must be maintained by both, either or neither the helo and ship. Additionally, the conditions that constitute a helo or a ship being in a support position, such as distance and orientation can be modified. Finally, the rules for when the boarding team needs to be supported can be established. In this scenario, the PCO must maintain support for the boarding team from either their helo or their ship once tensions aboard the suspect vessel have been elevated above a normal level. If the PCO adjusts their ship's course so that it can receive the helo, resulting in the ship not remaining in a support position

and recalls the helo, then the boarding crew would be left without adequate support. In this case, the PCO would receive a poor assessment as feedback from the system.

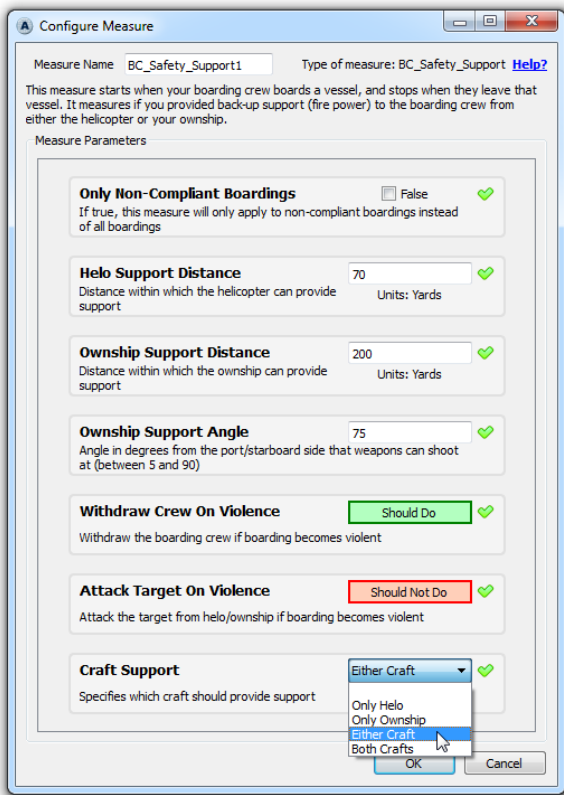


Figure 3. Measurement Configuration Dialog

In addition to using the performance measures to provide feedback to the students, the B2B training system uses them to guide the selection of scenarios. Scenario selection occurs dynamically. After each scenario, performance is measured and the next scenario is selected based on this measurement. The B2B training system requires three inputs in order to select scenarios. First, it requires a *measurement* model which it uses to construct an estimate of student *state*. Student state consists of a level of expertise for each different competency, which is being trained. Second, it requires a *transition* model. The transition model predicts the effect of each possible scenario on student state. Third, it requires a *utility* model. This specifies the training goals, e.g. it might specify which student competencies are most important. The B2B training system makes use of a POMDP (Partially Observable Markov Decision Process) model in order to perform scenario selection. It selects the scenario that is expected to best advance the student state in order to meet the training goals. Furthermore, it selects scenarios, which are part of scenario sequences. That is, the POMDP model looks ahead into the future, to

choose scenarios that lead into other scenarios that will accomplish the training goals.

By taking advantage of automated performance measurement, B2B is able to minimize instructor workload and time, and optimize the students time in the classroom. This is accomplished while providing the students with more objective and consistent feedback than could be provided directly from the instructor.

Tool Development

Design Planning and Tool Users

The ability to build new scenarios and to add new behaviors and procedures are critical for the long-term use of any training system. Every training system is (hopefully) useful the day it is delivered, but quickly becomes out of date as soon as either friendly or enemy forces change their tactics, techniques, and procedures (TTPs), new capabilities are added, or the situation otherwise changes. Ideally, the organization responsible for the training system (i.e., its “owner”) should be able to modify it to reflect these changes. If this is not possible, the owner should be able to contract with whatever organization it desires to make the changes. However, too often these changes can only be made by the developer of the application, which means that the owner is “locked-in” to that developer and has no other options for modifying the training application.

In order to give SWOS the ability to make the required changes to *Back to Basics*, the team developed tools for SWOS personnel to be able to modify as many aspects of the game as possible. Delta3D handled the overall tool design and implementation, while Aptima created plug-ins to work with-in these tools for creating and modifying performance measures. The goal was to give SWOS the ability to change as much as possible about the game and measures so that it was free from relying upon outside organizations as the game required future modifications.

Before beginning to build any of the tools necessary to do this, we first looked closely at the pipeline SWOS uses for development of scenarios for its current simulators. While not wanting to limit our product to current tools, we wanted our tools to fit in as seamlessly as possible to “business as usual.” Currently, scenarios are written using Microsoft Excel by the instructors at SWOS, generally lieutenants and lieutenant commanders. The Excel file is given to a group of civilian SWOS employees or contractors.

These civilians are highly technically savvy, and because they don't rotate as often as SWOS instructors do, they serve as the long-term institutional knowledge, especially in the programming of SWOS' simulators. They take the instructors' vision in Excel and enter it into the simulator so students can run it. This is the milieu in which our tools needed to fit.

Originally, we intended for only the instructors to use the tools to create scenarios and develop behaviors in the application. Because SWOS instructors are chosen for their performance at sea rather than their technical capabilities, we had to assume a low baseline for their computer skills: proficiency with most common applications, but nothing more. Additionally, we did not want them to have to spend significant amounts of time learning to use the tools we created, because too often such a tool remains unused and the training simulation is not updated and becomes useless. However, we soon realized that an interface simple enough for someone with those limited skills would not be powerful enough to modify and add behaviors as required.

We overcame this hurdle by deciding to build two tools for use by SWOS personnel. The first tool, the scenario builder, would be for the instructors' use and would be simple and intuitive enough that an instructor would be able to create and modify complex scenarios with little more than an hour's training. The second tool, the component builder, would be designed from the outset for the technical staff at SWOS. This would streamline the current process by allowing instructors to build scenarios without needing the SWOS tech staff, but would still allow SWOS to add and modify behaviors as they changed in the fleet. When it was proposed to both instructors and the technical staff at SWOS, they agreed it would work and fit nicely into their current workflow.

Figure 4 shows how this approach fits into the final deliverable product. Based upon the functionality that SWOS wanted the game to have, the Delta3D team built actions and AI behavior *B2B* needed to provide this functionality and Aptima built measures to evaluate the students' performance during scenarios. We delivered these items, in the blue oval, to SWOS as a beginning library of building blocks from which to construct scenarios as well as to serve as examples for SWOS in creating additional building blocks.

The Delta3D team designed the Component Builder tool for the SWOS technical staff to use to build additional basic elements for inclusion in the library. SWOS instructors then use the Scenario Builder to take these basic elements and produce scenarios with them.

If instructors require new components or measures because of a change in TTPs by either coalition forces or pirates, instructors will describe the changes to the tech staff, which will build them using the Component Builder and add them to the library for instructors to use in scenarios.

The Delta3D team created the both the Scenario Builder and Component Builder by modifying Delta3D's built-in design tools, STAGE. STAGE is a world-building tool which allows the user to place objects in the world and attach properties to them. One of the most-important features of STAGE is its plug-in architecture, which allows developers to modify it to fit the specific needs of a given application. The Delta3D team has used this aspect several times in the past to build the exact tool needed for the user, and has used it again in building both tools to give each group precisely the functionality each required.

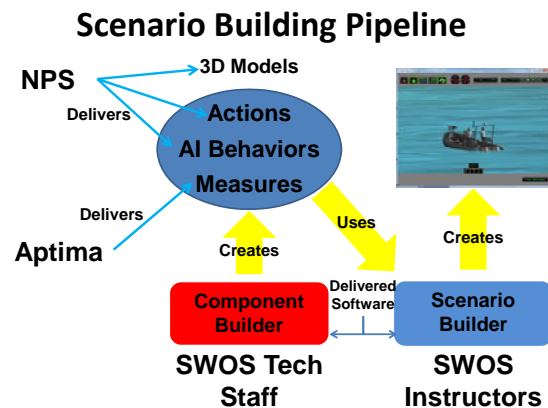


Figure 4. *Back to Basics* Scenario building pipeline

Scenario Builder

The Delta3D team designed Scenario Builder to give instructors an intuitive, easy-to-use tool to build complex scenarios quickly. It is much simpler than either STAGE or the Component Builder – modifying STAGE to create Scenario Builder involved mainly removing functionality and making other aspects easier to use. It gives instructors the ability to perform all the tasks they need to create a scenario, including:

- Placing vessels in the world;
- Choosing the appearance and physical characteristics of vessels;
- Attaching behaviors to vessels;
- Adding crew members, along with their behaviors, to vessels;

- Setting time of day and weather/sea conditions;
- Creating the “current events” to set the scene the scenario is taking place;
- Attaching metrics that inform the performance measure system of the student’s performance in the scenario.

By giving the instructors this functionality, we have simplified the SWOS scenario creation process. No longer do instructors need to ask the technical staff to create scenarios; instead, the technical staff is only needed to create new procedures using the Component Builder.

Component Builder

Because the desired functionality was significantly more complex than the Scenario Builder, Component Builder was more complicated to design and build than the Scenario Builder. While the basis of Component Builder was still STAGE, it also used two of STAGE’s more advanced plug-ins, one which existed previously and one which was created for this project.

The Delta3D team used STAGE’s Director functionality, shown in Figure 5, for a tool which gives the technical staff to create scripted behaviors. Director is a node-based graphical programming environment which allows non-programmers to create complex events. For example, in Director the user can create a string of events and attach them to a trigger volume placed in STAGE. When a game entity enters the trigger volume, the string of events commences. Similarly, events created in Director can be invoked many ways, such as an event occurring in the game world or the player performing some action.

In the Component Builder, the tech staff uses Director to create procedures, such as what a group of suspect vessels does when approached by a Navy ship. The staff can create one where the suspects are very compliant, another where they ignore the communications from the Navy ship, another where they flee together, and another where they all flee in separate directions. The tech staff then adds these events to the building block library, and the instructor chooses which of them he wants the suspects to perform in the scenario and attaches it to the suspect vessels in Scenario Builder. Once the tech staff has built the events, instructors can use the same events in many different scenarios without requiring any further tech staff interaction.

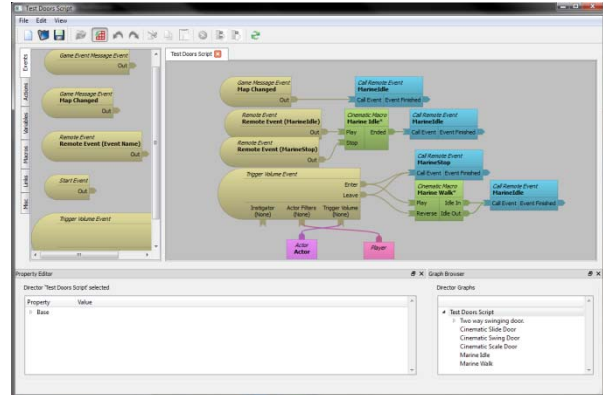


Figure 5. Delta3D Director

Director was used to create simple event strings, but in some areas the game required actions more subtle than users could build in Director. We called these actions “behaviors” to reflect that they are more closely associated with AI. Delta3D has long had an AI system based upon Jeff Orkin’s AI in F.E.A.R. (Orkin, 2006). This is essentially an AI planning algorithm adapted for games. Planning algorithms have actions which change the world state and may be called only if certain conditions, called prerequisites, are met. Each action changes the world state, called final conditions. The planning algorithm sees if it is possible to change the world state from the existing state to the goal state by chaining together actions. (For more information on planning algorithms, see Russell & Norvig (2002)).

However, Delta3D’s planning algorithm was completely programming based and as such only programmers could utilize it, which was unacceptable for the tool we needed to deliver to SWOS. While the SWOS technical staff is highly competent, their areas of expertise do not include C++ programming and we could not expect them to perform easily tasks requiring programmers. Therefore, while the AI system provided the functionality we needed to create behaviors, we needed to create an interface for non-programmers.

Since the tech staff was already using a Director based tool, the Delta3D team decided to create an interface very similar to Director’s node based system to allow non-programmers to use the existing SI system. In this system, users add prerequisites and nodes have and final conditions to nodes, which represent actions. This makes it very easy for the tech staff to create advanced behaviors without scripting.

Integration

The greatest technical and schedule risks in developing the B2B game were due to multiple integration points between components developed by Delta3D and Aptima. To mitigate these risks Delta3D and Aptima held weekly technical meetings, aligned development schedules and front loaded the integration work.

In B2B, Delta3D and Aptima's performance measurement software interact during both scenario authoring and the runtime of the game. Therefore the integration between Delta3D and Aptima's performance measurement software took place within STAGE, the Delta3D scenario-scripting tool as well as within the Delta3D game engine. For the first point of integration, Aptima developed a new piece of software, which allows users of STAGE to add and configure, measures for given scenarios. Aptima developed this measurement configuration component as a plug-in for STAGE, which enables the users to configure performance measures while they are laying out the scenario without having to use multiple tools. For the Delta3D runtime integration, Aptima provided the Delta3D team with a specification for sending data to the PM Engine. This specification included both the transport protocol and the message structure. The Delta3D team then iteratively enabled the required game data to be published according to the specification.

In order to better support each other during development, Delta3D and Aptima aligned their development process and worked in regular two week sprints. Additionally, Delta3D and Aptima made the deliberate decision to complete the integration as early as possible. In the first sprint, Aptima developed a preliminary version of the Stage plug-in which, when used, resulted in valid measures described in HPML being stored at a specific location within the scenario files. During the second sprint, Delta3D enabled the Delta3D engine to both send the previously generated HPML to the PM-Engine when a scenario is loaded as well as send regular updates of entity and environmental states and simulation events as the game is played.

By holding regular technical meetings, aligning development schedules and completing integration in the early stages of development, Delta3D and Aptima were able to minimize risk and foster a collaborative development process. This ultimately led to an efficient development process and an integrated product.

Conclusion

The *Back to Basics* product developed for SWOS provides an example of how a game environment can be integrated with performance assessment technology to provide training experiences in a complex domain where there are no easy answers. The training product was produced relatively quickly, with limited resources, and was designed to run on the existing technology infrastructure in place at SWOS. The game environment provided the ability to quickly create displays and scenarios, and supported rapid authoring capabilities. The automated performance assessment technology, coupled with technology to select subsequent experiences based on the student's performance, allowed the game to function as an intelligent tutoring system, making effective use of both limited instructor availability and limited student class time.

ACKNOWLEDGEMENTS

The authors would like to thank ONR's TechSolutions branch for their sponsorship and help on this project. Additionally, we would like to thank personnel at SWOS who were so giving of their time to help us understand the domain and what they desired as a final product.

REFERENCES

- Beach, E.L. (1986). *The United States Navy*, New York: Henry Holt and Company, pp24-27.
- Orkin, J. (2006). Three states and a plan: The A.I. of F.E.A.R. Presented at the 2006 Game Developers Conference, San Francisco, CA. Retrieved 6/19/11, from http://web.media.mit.edu/~jorkin/gdc2006_orkin_jef_f_fear.pdf,
- Russell, S., & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach* (2nd Edition). New York: Prentice Hall.
- Surface Warfare Officer School, (2010). Lesson Plan for Unit 1/Topic 5: Maritime Interception Operations (MIO) Task Force Structure, Tactics, Communications, & Rules of Engagement, Newport, RI: Surface Warfare Officer School, page 1.
- The Deadly Years. (n.d.). In *Wikipedia*. Retrieved August 19, 2011, from http://en.wikipedia.org/wiki/The_Deadly_Years