Use of Automated Intelligent Entities in ASW Simulation

Morten Kolve
Kongsberg Defence Systems
Kongsberg, Norway
morten.kolve@kongsberg.com

Jared Snyder
Discovery Machine Inc.
Williamsport, Pa, USA
jsnyder@discoverymachine.com

Geoff Tompson
Decisive Encounters Limited
London, UK
g.tompson@de-ltd.com

ABSTRACT

As defense budgets are cut, assets and personnel are being increasingly stretched to meet operational tasks, making it ever more difficult to allocate platforms and subject matter experts (SMEs) to train the next generation of operators. One key area where there is a shortage of platforms and experienced SMEs is in anti-submarine warfare (ASW). At the very time when many countries are purchasing sophisticated submarines and the potential submarine threat is increasing, fewer operational submarines and SMEs are available for training tasks. An innovative solution to these shortages is to use an 'expert system' ASW simulator employing automated intelligent entities to generate realistic threat actions. This innovative solution provides added benefits as it improves the quality of simulator training whilst reducing the workload on the available ASW simulator instructors. This Paper describes the process used for collecting expert knowledge and then using that knowledge to create the automated intelligent behaviors employed by automated intelligent entities in simulators. The collection process enables SMEs to ensure the behaviors represent tactically realistic actions and, for the highest quality simulation, that they do not become predictable. Therefore, for any tactical situation, the system must select the most appropriate behavior and the entity should react realistically to the tactics employed against it by the student. Such autonomous entities allow instructors to perform complex maneuvers and actions with a low level of interaction with the simulation. An additional benefit of the low level of interaction with the simulation is the reduction in the instructor's workload, giving them more time to focus on the overall simulator exercise objectives. As an illustrative example, we present a case study of a system created for the Royal Norwegian Navy (RNoN), which now uses such automated intelligent behaviors in its ASW simulator.

ABOUT THE AUTHORS

Morten Kolve is from Norway and is currently the Marketing and Business Development Manager for navy simulators and trainers at Kongsberg Defence Systems, a supplier of a wide range of military systems. His interest and work for more than twelve years has been systems design and development of naval team and task group level simulators, with emphasis on enhancing team and task group level performance and coherency through new simulation capabilities. Prior to his current work in simulation and training, he worked for eight years as a systems engineer on naval combat management systems. He also served for ten years in the RNoN as an underwater weapon systems officer and holds an engineering degree in computer science and electronics from Gjøvik University College.

Jared K. Snyder is from the USA and is currently a Senior Software Engineer for Discovery Machine Inc., where his work focuses on behavior modeling and developing core technology offerings. After graduating with honors from Bucknell University with a Bachelor's of Science degree in Computer Science, he has been designing and developing software and behavior models for Discovery Machine Inc. since 2009. His work has focused on creating behavior models for simulated entities that behave and interact in a human-like manner within constructive military training environments.

Geoff Tompson is from the United Kingdom (UK) and served for 20 years in the UK's Royal Air Force and Royal Auxiliary Air Force, including more than 7 years flying maritime patrol aircraft (MPA) as a Navigator and Captain. He flew MPA at the peak of the Soviet submarine force's capabilities and accrued many live hours of tracking transiting and on-station submarines. Since retiring from the RAF, he has been involved in military training, including spending 6 years teaching in the Middle East, initially as an Electronic Warfare Officer, and subsequently as an adviser at a Naval Forces School with specific responsibility for the use of simulation. For the past 14 years, he has been involved in the application of the 3D modeling of environments and military platforms for use in a number of training and simulation products that are now in service with more than 25 users worldwide. As the CEO of Decisive Encounters Limited, he is currently working with Kongsberg Defence Systems to develop new naval training systems.

Use of Automated Intelligent Entities in ASW Simulation

Morten Kolve
Kongsberg Defence Systems
Kongsberg, Norway
morten.kolve@kongsberg.com

Jared Snyder
Discovery Machine Inc.
Williamsport, Pa, USA
jsnyder@discoverymachine.com

Geoff Tompson
Decisive Encounters Limited
London, UK
g.tompson@de-ltd.com

INTRODUCTION

As defense budgets are cut, assets and personnel are being increasingly stretched to meet operational tasks, making it ever more difficult to allocate platforms and subject matter experts (SMEs) to train the next generation of operators. One key area where there is a shortage of platforms and experienced SMEs is in anti-submarine warfare (ASW). At the very time when many countries are purchasing sophisticated submarines and the potential submarine threat is increasing, fewer operational submarines and SMEs are available for training tasks. An innovative solution to these shortages is to use an 'expert system' ASW simulator employing automated intelligent entities to generate realistic threat actions to provide the following:

- Realistic threats.
- Meaningful threat reactions to student actions, tactics and the environment.
- Reduced instructor workload.
- Reduction in 'skill fade'.

A potential weakness of an 'expert system' is that it may only have a limited repertoire of actions and reactions. After only a relatively few exercises, trainees may learn the full extent of the system's repertoire and concentrate on 'beating the system'. In reality, the real world can often be predictable too. It is not just Commander Ramius in 'The Hunt for Red October' (Clancy, 1984) who had his favorite stern arc clearance maneuver. When tracking submarines for days on end, we often found they fell into a routine, perhaps through complacency or maybe they had been trained by a poorly-engineered 'expert system' offering limited options. Unlike them, we were tracking many of their contemporaries so after a while we learnt most, if not all, of the various Captains' favorite schedules and maneuvers. When developing our 'expert system' we therefore decided it needed to have a broad repertoire.

Another weakness of some 'expert systems' is that the software reacts too quickly to user inputs, leaving them frustrated that the enemy is always one step ahead. In the real world, the submarine may not know it is being tracked so cannot anticipate when to react, perhaps to a pattern of sonobuoys being dropped ahead of it. Also, with real world systems, there is a finite time between an event being detectable and it being detected, perhaps due to data processing time or maybe the sensor operator's alertness on that occasion. We therefore wanted our 'expert system' to act and react in a timely fashion and not necessarily react immediately to user actions. There would be exceptions of course, such as the immediate reaction to an ASW helicopter going active with its sonar or radar. ASW is a 'game of patience' and we wanted our users to learn to be methodical in their actions and to be patient. If they chose to go active too soon, they needed to learn by their mistake.

This Paper describes how a Royal Norwegian Navy (RNoN) ASW simulator was adapted to meet its training needs. It describes the process used for collecting expert knowledge and how that knowledge was used to create the automated intelligent behaviors employed by the automated intelligent entities inserted into the existing RNoN simulator. Underpinning the trial was the need to create a realistic 'expert system' whilst utilizing the RNoN's existing ASW simulator adapted using off-the-shelf third party middleware software.

ASW SIMULATION CHALLENGES

Tactical simulators are used to train naval operators in skills and competences ranging from basic war fighting skills, such as communication procedures and plotting, through team building and decision making in the ship's Combat Information Centre (CIC), all the way up to commanding integrated task groups. Based on the specific training

objective of individual exercises, students may train alone or participate as a member of a team. To create an immersive training environment, students will normally be located in a simulated naval platform within a realistic synthetic environment representing the real world - 'ground truth'. The instructor will have planned the exercise based on the training objective(s), with emphasis on the best pedagogic approach to ensure learning. The supporting scenario will be kept as simple as possible, both to maintain focus on the objective(s), and to ensure pedagogic predictability for the instructor.

As training objectives become more complex, so do the required simulation scenarios, often involving many more entities. During scenario preparation the instructor can script non-student controlled entities behavior over time, such as start position, initial course and speed, the times of course changes and of active emissions. The script will usually be fixed to the anticipated timeline and synthetic environment. During this preparation phase, instructors may also assign basic reactive behaviors to the non-student entities. Scripting reduces the need for instructor control and micro-management of the non-student entities during the simulation exercise. However, there is still a need for the instructor to continually monitor the script to ensure it remains relevant to the training objectives. For example, the actions of student-controlled entities could impact on the script, perhaps in a simple way such as requiring the instructor to maneuver a traffic ship in accordance with the Rules of the Road. So, despite scripting simple tasks, the need to monitor all entities would continue to be a major task for simulator instructors and could potentially distract them from monitoring student performance against the training objectives. To reduce the risk of distracting the SME instructor(s), role-players are often employed to interact with the students, as either hostile or friendly entities. Role-players may be other instructors, personnel specifically employed as role-players, or personnel temporarily assigned to the school when not required for sea duty. However, such role-players may not have subject matter expertise for the role they are required to play in the exercise, so the instructor must monitor their actions, which to some extent defeats the objective of having role-player assistance to reduce instructor workload.

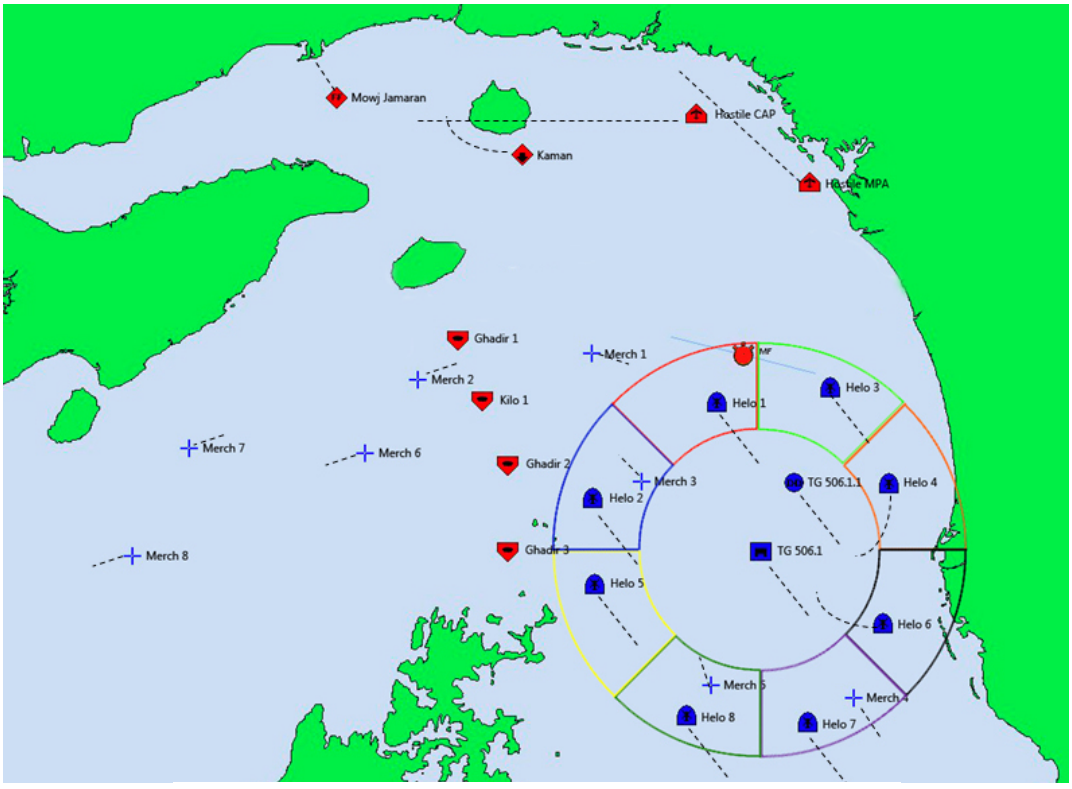

Figure 1. ASW Task Group Transiting the Strait of Hormuz

Figure 1 represents a typical complex scenario in the confined waters of the Strait of Hormuz, where an ASW Task Group is transiting into the Persian Gulf opposed by 4 submarines. During scripting, non-student vessels transiting through the Strait will maneuver at appropriate times to follow the designated traffic separation scheme (TSS).

However, if a student entity crosses one of the TSS channels, it may be necessary for the instructor or a role-player to maneuver the affected non-student vessel to avoid a collision.

Providing realistic simulation training to students in such complex naval scenarios requires a significant staff of instructors and role-players. In many cases, SMEs have to split their time between monitoring the students, role-playing and interacting with the other instructors. However, to provide the optimum quality of training, instructors need sufficient time for coaching students and providing immediate feedback to them on their actions.

The RNoN was initially attracted to employing artificial intelligence (AI) to control some entities in its naval simulators having observed the use of AI in JSAF (Joint Semi-Automated Forces) exercises. Operational commitments were making it increasingly difficult to release SME instructors and operational submarines for live training, which focused the RNoN's training school staff on finding ways to optimize their simulators. Their aims were to both reduce workload on the available SME instructors, and to minimize the need for live training against friendly operational submarines. Their initial interest prompted the subsequent practical solution of using automated intelligent entities controlled by automated intelligent behaviors in their simulators. In early 2013, we proposed conducting a prototype Research and Development (R&D) level trial utilizing an AI system already proven in JSAF, integrated into the RNoN's synthetic ASW training environments. While the main driving force for the trial was to reduce instructor workload during simulator exercises, the integration gave the added benefit of turning the existing ASW simulator into an expert system providing a highly realistic operational environment, which reduced the need for using friendly operational submarines in live training.

AI SYSTEM USED

The AI system used was a COTS (commercial off-the-shelf) simulator-agnostic middleware system which could be quickly integrated with the existing RNoN simulator with minimal changes to either the middleware or the RNoN's simulator. We felt this provided minimum risk, would speed up integration to allow the R&D integration to proceed with minimum delay, and would therefore save money.

This middleware system – the Behavior Creation Toolkit (BCT) – was used to perform the following operations:

- Capture SME knowledge via a modifiable console
- Store reusable captured knowledge in a Knowledge Service Library
- Execute compilations of captured knowledge with a Knowledge Service Engine
- Provide access to higher level functionality such as Text/Audio Communications

The console portion of the BCT framework facilitated the collection and representation of expert knowledge. The process used to capture knowledge is explained in a later section. The integration of the BCT with the RNoN ASW simulator required specification of a thin integration layer comprised of the following components:

- **Action functions** – functions called by the behavior model to cause the entity to perform actions in the simulated environment, such as movement to a particular location, turning sensors on and off, or firing a weapon.
- **Query functions** – functions called by the behavior model to request information about the simulation, such as the current state of the entity driven by the behavior model, or what sensory stimuli are currently being observed by it. The key differentiator between queries and actions is that queries expect a response from the simulation.
- **Data structure specifications** – a specification of data types passed back and forth between the simulation and the behavior models, used by the functions described below. (Potts et al, 2010)

In the case of the RNoN simulator, the following function set was defined for the BCT based upon the set of SME knowledge to be captured:

- deployDippingSonar(EntityID, DippingSonarData)

- deploySonobuoy(EntityID, DeploySonobuoyData)
- getContacts(EntityID, EntityID)
- getEntityState(EntityID, EntityID)
- getNearbyVessels(EntityID, NearbyVesselRequest)
- getRawBathymetryData(EntityID, LatLonLocationData)
- getSimTime(EntityID)
- setDirectionSpeedAltitude(EntityID, DirectionSpeedAndAltitudeData)
- setFireTorpedoData(EntityID, FireTorpedoData)
- setPeriscopeOperations(EntityID, PeriscopeOperationsData)
- setSensorData(EntityID, SetSensorData)

The behavior models created with the Behavior Creation Console (Figure 2) leveraged these functions as low-level atomic actions to directly interact with the specific simulation environment. Once the functions set and data structure were created, which converted the BCT from being simulation-agnostic to simulation specific, the actual knowledge capture began.

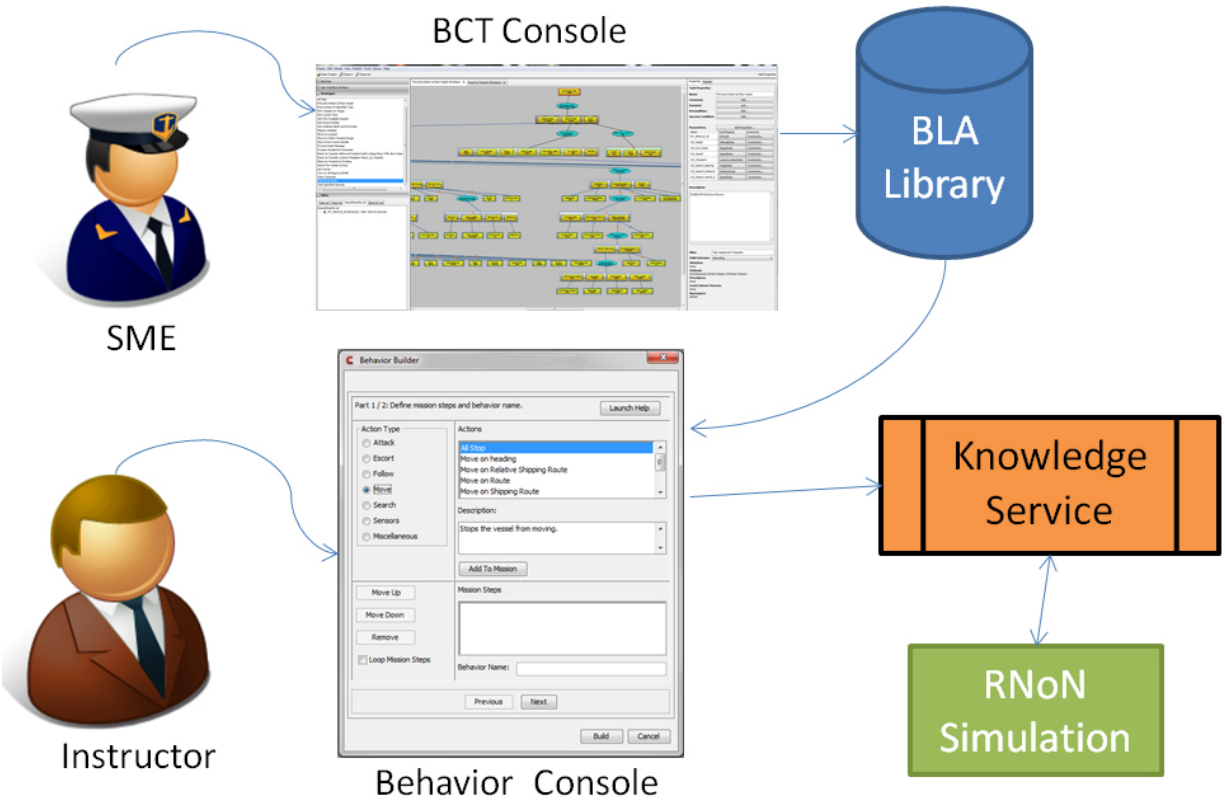

Figure 2. Process and Tools Used to Create Behaviors

PROCESS FOR COLLECTING EXPERT KNOWLEDGE

Capturing expert knowledge in the context of building our 'expert system' focused heavily on the creation of Basic Level Actions (BLAs), which were built using a visual programming language called Task Method Knowledge (TMK). Each BLA encapsulated an independent operational module containing the SME's experience of the steps they would take to perform a specific operational task. After identifying and creating a set of BLAs, which comprised a subset of tactics used in a specific type of warfare, in this case ASW, we used them to create behaviors for the automated entities. By combining the created independent operational modules, we were able to create a

large set of ASW behaviors. We used a process to capture the SME's knowledge based on a proven methodology that facilitated introspection and articulation of subject matter expertise (Potts et al, 2010). This methodology for knowledge capture allowed the design process for our system to proceed from highly abstract levels of representation to realistic operational levels. The methodology consisted of the following phases:

- The initial phase consisted of a structured interview process with the SME. The structure within the interview encouraged the SME to think about problem solving within the BLA from many points of view, which uncovered the tacit knowledge that made our SME an expert in their particular field.
- BLAs were constructed rapidly using the TMK modeling software resulting in their logic being both transparent and executable. The TMK's visual representation of the BLA's inherent structure was easy to understand (for checking by the SME) since each section of the hierarchy was autonomous and understandable within itself.
- Having constructed the BLAs, they were each tested to ensure they performed correctly.
- Finally, the full library of BLAs was compared against the knowledge captured from our SMEs to ensure the library met the RNoN's objectives of reducing simulator instructor workload and creating an expert system to improve the overall quality of training provided.

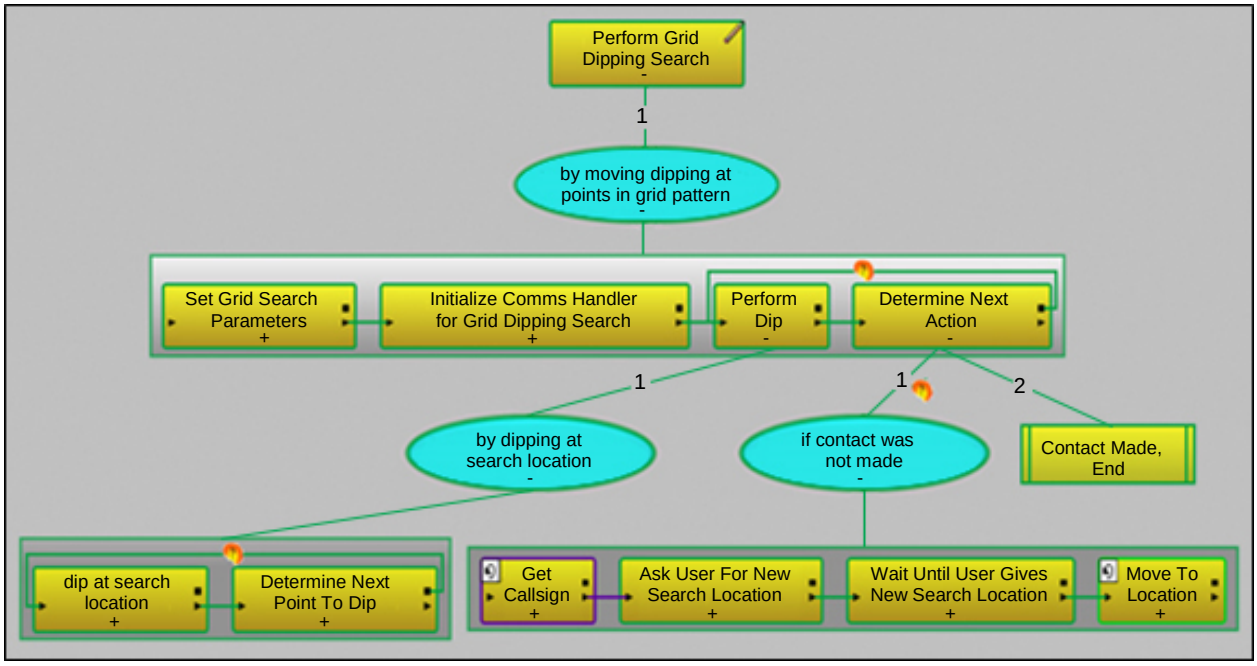

Figure 3. Perform Grid Dipping Search BLA

As an example, the ***Perform Grid Dipping Search*** BLA (Figure 3) was created as follows:

- A RNoN ASW instructor SME was interviewed by a trained engineer who ascertained that performing a subsurface search using a helicopter's dipping sonar was a common ASW training requirement.
- Having identified this training need, the trained engineer was able to elicit the nuances in how this action could be performed and what environmental factors may cause the behavior to adapt its tactics. While the SME explained how they performed this training task, the trained engineer created a structure using TMK.
- This initial behavior model was shown to the SME, who provided feedback on the quality of the BLA to the engineer. During this process, further reflection occurred and any missing actions or corrections were added.
- After several iterations of this process, the SME was able to validate the created structure.
- Having created a validated structure for the BLA, the engineer added the finishing touches to allow it to interact with the RNoN's simulation.
- The validated ***Perform Grid Dipping Search*** BLA was then added to the BLA library.

Once the engineer had created a library of BLAs, they required additional structures to make the automated intelligent entities adaptive to changes. These additional structures were situational awareness (SA) and reactive behaviors, which were needed to make the automated entities adaptive enough to truly assist the simulator instructors.

CREATING AUTOMATED INTELLIGENT BEHAVIORS

In order to create automated intelligent behaviors that alleviated the need for RNoN instructors to micromanage them, three goals had to be achieved:

- Instructors needed to trust the tactics used
- Small adaptations would be made automatically based on environmental changes
- Entities would react to critical changes in their environment

The first goal was largely accomplished by using the SME knowledge encapsulated in the BLAs. In order to satisfy the other goals, an automated entity required situational awareness to be able to recognize important changes in the environment and be reactive to those observed changes.

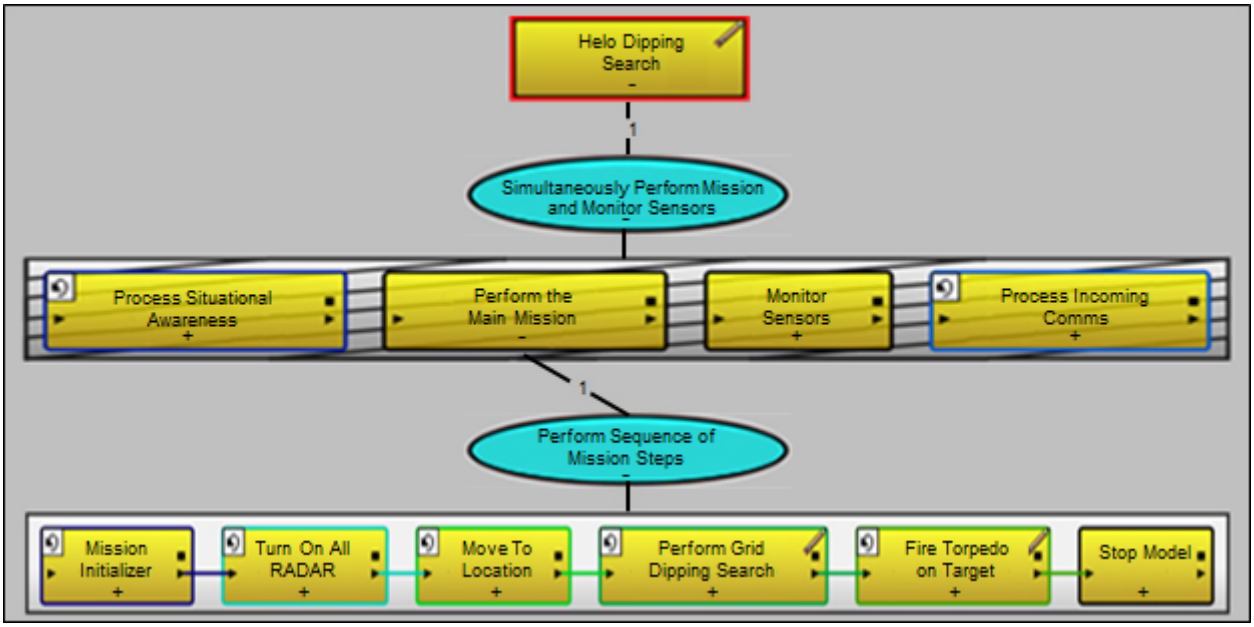

Figure 4. Example Automated Intelligent Behavior

Automated entities that are aware of their surroundings make better decisions. As such, the behaviors built in the RNoN case study had a Situational Awareness Module (SAM) that processed low level data to create a higher level abstract understanding of the entity's environment. The SAM consisted of a series of subtasks repeatedly executed in sequence throughout the lifetime of the behavior. Each individual task processed some piece of SA data. For example, all raw sensory information was collected and stored in the SAM. This raw data was then processed into a concise list of contacts, with all of the sensory information being fused into a single, higher level data structure. This assisted the BLA building process by allowing the RNoN's SMEs to think at the level of contacts and not at the level of raw sensory data. This also provided the behavior model with an understanding of the simulated environment, an "interpreted" world view instead of simply ground truth. Two benefits of using this "interpreted" world view in the BLA's decision making process became evident: firstly, the complex cognitive modeling that took place in processing the raw data into useful concepts was performed outside of the BLA. Secondly, since the world view was derived from the raw sensory information, it ensured the automated entity behaved in a way that was in accordance with what it believed the world around it looked like.

Another of the subtasks in the SAM was designed to observe critical changes in the state of the environment. In the case of ASW, these could include: detecting an active ping hitting the controlled submarine; detecting a torpedo in the water; a collision with another ship or land and so forth. When one of these alterations in the environment was observed, the SAM sent the notifications required to initiate reactive measures. In some cases this involved interrupting an entity's current behavior in order to avoid a catastrophic event, such as an approaching collision or weapon, or notifying the currently executing BLA so that it adjusted its tactics based on the new information. In some cases, the automated intelligent entity found it necessary to change its entire behavior to another more appropriate course of action, for example, if an escalation of force occurred. In this way, the same behavior adapted and reacted to a large set of possible courses of action. This was in contrast to following a script, which could have had problems dealing with a trainee taking an unexpected action. The following case study will describe some examples of the adaptability of this AI approach.

CASE STUDY

Having observed and analyzed the integration of intelligent behaviors into the US Navy's JSAF, we approached the RNoN to conduct a prototype R&D level integration into the synthetic environment they used for tactical training. The objective of the prototype R&D project was to investigate if the technology used in JSAF could be utilized to assist the RNoN's simulator instructors by automatically ensuring realistic behaviors for single entities, and to perform routine tasks such as voice reporting. The primary objective was to reduce instructor workload using automated intelligent behaviors so their attention could be focused on the trainees' actions. We also hoped that achieving this objective would reduce the need for role-players.

ASW is a RNoN focus area so was selected as the case topic for the prototype project. However, ASW is a very broad subject so we focused on a specific case: a hostile submarine being prosecuted by a friendly ASW task force. The trainees would participate as a frigate's ASW warfare officer and his/her team in the CIC. Automated intelligent entities were to play the friendly airborne ASW assets (maritime patrol aircraft (MPA) and ASW helicopters) and the hostile submarine, roles that were previously played by various SMEs.

The initial trial included basic integration between a RNoN simulator and a middleware system that controlled the automated entities' behavior simulation. The middleware then executed the assigned behavior and updated the entities in the synthetic environments accordingly. Thus, for the trainees, the behavior of the entities appeared as normal, or even better, given that its modeled SME behavior controlled them.

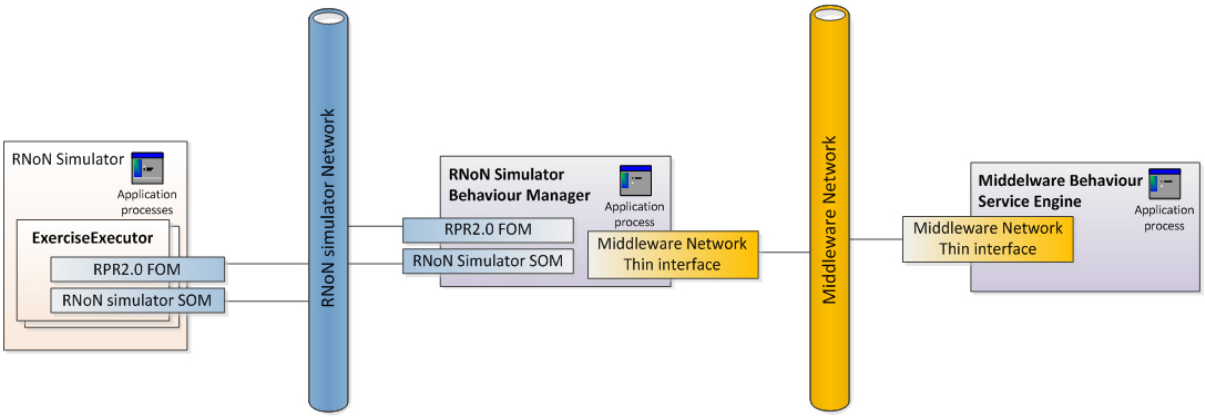

Figure 5. Integration of the RNoN Synthetic Environment with the Middleware's Behavior Service Engine

The first step of the integration was to connect the synthetic environment in the RNoN's ASW simulator with the middleware's service engine. We achieved this by introducing an executable into the RNoN simulator called BehaviorManager that acted as a gateway to exchange messages between the simulator and middleware (Figure 5). The application ticked the middleware's Behavior Service Engine and received commands. Commands requiring a response were actioned and an appropriate response transmitted back. Technically, the BehaviorManager linked directly with the middleware and implemented a set of functions derived from the requirements of the RNoN's ASW

simulator via its 'thin interface' API (Application Program Interface). A TCP (Transmission Control Protocol) connection was created from the BehaviorManager application to the middleware's Behavior Service Engine, which was running on the same computer, whilst Google Protocol Buffers were used to send messages back and forth between the middleware's Behavior Service Engine and the BehaviorManager.

Having linked the RNoN simulator to the Behavior Service Engine we collected the appropriate expert knowledge, created the BLAs and used them to create the automated intelligent behaviors, as described earlier in this Paper.

Result

Having integrated the Middleware Behavior Service Engine with the RNoN simulator the system was tested to ensure the original design objectives had been met. A key concern for any operator is the need to avoid the behaviors becoming predictable as this would undermine the quality of the training. Certain processes, such as immediate action drills (IADs), allow little scope for variation. However, once a target had been detected and IADs carried out, it was important that subsequent automated intelligent behaviors were both logical and unpredictable to constantly challenge the trainees. From an early stage, it was apparent this aim had been met and is best demonstrated using an actual example where 2 runs with the same start conditions resulted in very different but equally logical outcomes.

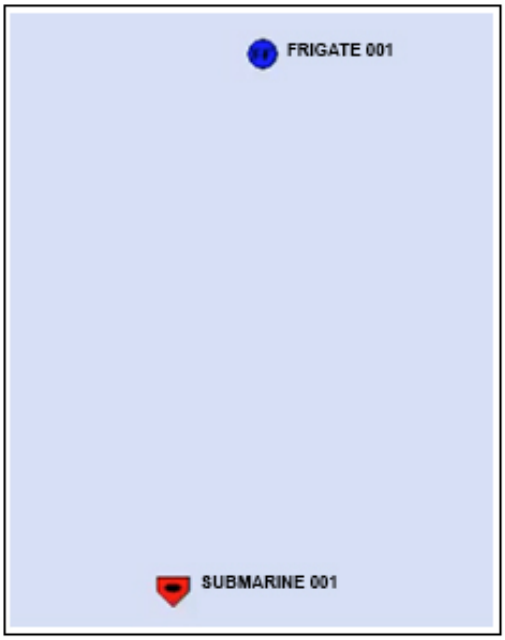

Figure 6. Time 0:00

Example Exercise Historical Plots

In the following example exercise, a student-controlled ASW frigate started to the north of a hostile submarine. The ASW frigate was on a course of about 190˚, and had just detected the submarine dead ahead; initially the hostile submarine was stationary. In the 2 runs to be described, the IAD and subsequent maneuvers carried out by each of the entities was different which, as will be seen, resulted in different outcomes. We describe events at significant times during the runs relative to start time in minutes and seconds (with 0:00 being the start time). The start positions of the frigate and hostile submarine are shown in Figure 6.

The screen grabs shown in Figure 7 were taken at time 3:20. In Run 1, the frigate's IAD was to turn to port to bring its starboard ASW torpedo launcher to bear and fire before continuing to turn away from the threat. In Run 2, the frigate's IAD was to turn to starboard, a torpedo was fired from the port launcher and the turn continued away from the threat. Meanwhile in Run 1, the submarine initially ran towards the incoming torpedo at maximum submerged speed and then deployed a decoy before turning to port to escape the torpedo's search box. Whilst in Run 2, the submarine turned to port away from the torpedo, releasing a decoy during its turn and increasing speed. At time 3:20 in Run 1, we can see that the submarine successfully escaped the torpedo's search box, whilst in Run 2, the torpedo was heading towards the decoy with the submarine escaping to the southeast.

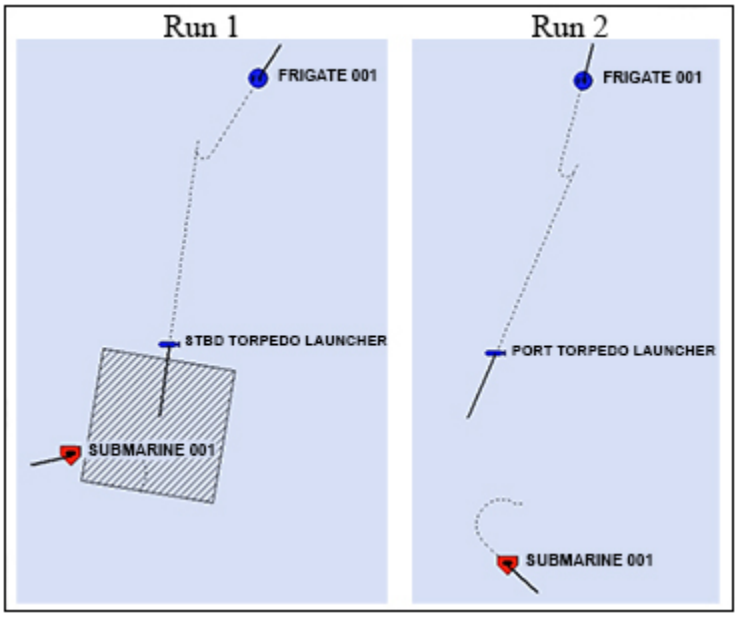

Figure 7. Time 3:20

The screen grabs shown in Figure 8 were taken at time 6:00. In both runs, the student-controlled ASW frigate turned back towards the threat to determine if its attack had been successful. However, the torpedo in each run had been seduced by the decoy released by the associated submarine, and both submarines had successfully evaded the torpedoes fired towards them. Having determined their attacks had been unsuccessful, the student-controlled ASW frigates commenced lost contact procedures and were attempting to re-locate the target submarines. Meanwhile, the submarines continued running out from the areas where they were last detected to make good their escape. As the next screen grabs demonstrate, both hostile submarines subsequently turned back into the fight.

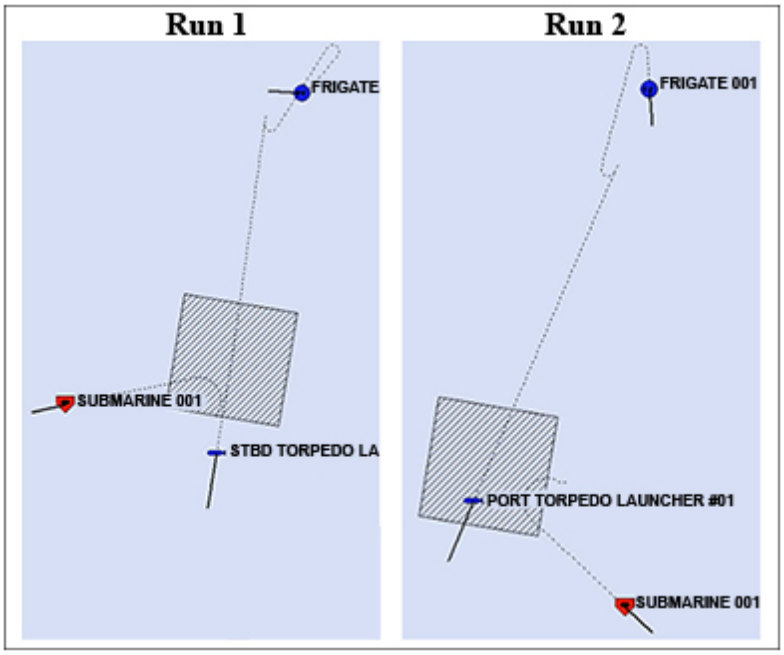

Figure 8. Time 6:00

The screen grabs shown in Figure 9 were taken at time 24:32. In both runs, the submarines went onto the offensive, with the submarine in Run 1 having just fired a torpedo. In each case, the submarine was approaching the frigate using a zig-zag course. However, it is noteworthy that the behavior adopted by the submarine in Run 1 was an asymmetric zig-zag path, whilst submarine 2 was following a more symmetrical zig-zag.

There is not enough space available in this paper to follow these example runs further. Suffice to say that, having started with the same parameters at time 0:00, the entities then displayed different but equally logical behaviors with no need for instructor input. Both submarines were initially put on the defensive as they evaded the incoming torpedoes then returned to the fight.

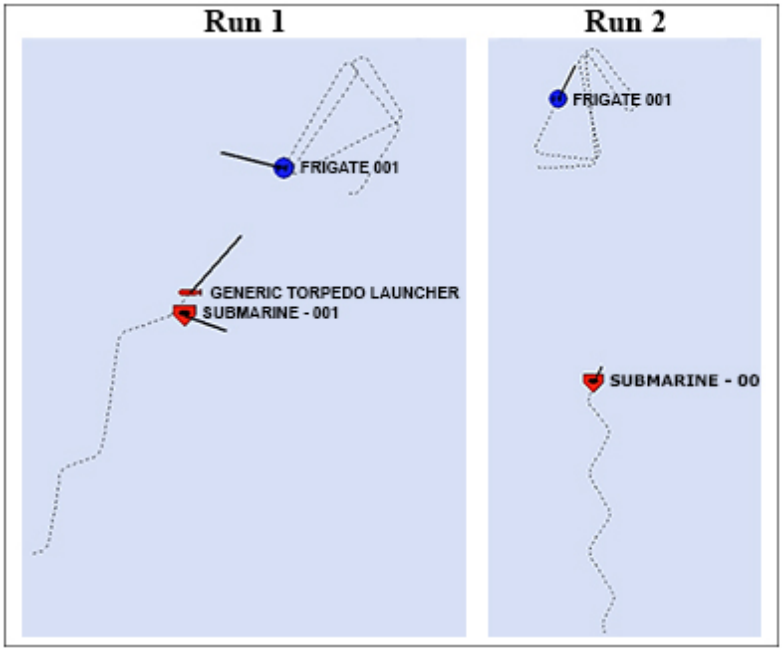

Figure 9. Time 24:32

WAY AHEAD

The scope of the initial trial was limited to collecting BLAs and creating automated intelligent behaviors to control an enemy submarine, friendly MPA and ASW helicopters in an existing RNoN ASW simulator. Having proven that the modifications made to the RNoN simulator did reduce both instructor workload and the need for an operational submarine to support training, the next step will be to capture more SME expertise to further reduce the reliance and workload on the RNoN's limited number of experienced ASW simulator instructors.

CONCLUSION

As defense budgets are squeezed but with no reduction in the operational tempo, it becomes harder for navies to release operational submarines for live training at sea and SMEs for instructional duties. This results in the few SMEs at training establishments being too heavily utilized, which can impact on the quality of training they provide. To alleviate this problem, role-players may be used during simulator exercises to supplement the available SMEs. However, role-players may not be familiar with all aspects of the training so SMEs must still monitor their actions whilst also monitoring their students. To overcome this vicious circle, the RNoN considered using AI to control intelligent entities to both reduce SMEs' workload and improve the quality of simulator training to offset a reduction of live training opportunities at sea.

To prove the feasibility of using intelligent entities, a prototype R&D project was carried out using an existing RNoN ASW simulator linked to middleware AI software already proven in JSAF. The R&D project focused on enabling friendly ASW assets to prosecute a hostile submarine. The RNoN simulator was linked via a bespoke interface called BehaviorManager to the AI middleware software. RNoN SMEs then assisted engineers who collected the appropriate expert knowledge, created BLAs (Basic Level Actions) and used them to create automated intelligent behaviors. Subsequent trial runs showed that the entities controlled by the automated intelligent behaviors displayed different, but equally logical behaviors, with no need for instructor input. The trial allayed fears that the automated intelligent entities would make the training too predictable. The next step will be to capture more SME expertise to widen the scope of training provided using automated intelligent entities. Thus our aim is to further reduce the workload on SME instructors whilst also reducing the need to release operational submarines for live training at sea.

ACKNOWLEDGEMENTS

Our thanks to Scott Hooper for his patience as our deadline approached.

REFERENCES

Clancy, T. (1984). Book Title: *The Hunt for Red October*. Location: Naval Institute Press

Franklin, F., Graesser, A., (1996). Proceedings Title: *Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents.* Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages. Location: Springer-Verlag. Berlin, Germany.

John, B.E., Prevas, K., Salvucci, D.D., Koedinger, K. (2004). Proceedings Title: *Predictive human performance modelling made easy*. Proceedings of CHI 2004 (Vienna, Austria, April 2004). Location: ACM, New York, USA.

Lebiere, C., Archer, R., Warwick, W. & Schunk, D. (2005). Proceedings Title: *Integrating modeling and simulation into a general purpose tool.* Proceedings of the 11th International Conference on Human-Computer Interaction. Las Vegas, NV

Nieh, J., Jae Yang, S., Novik, N. (2000). Technical Report CUCS-022-00 Title: *A Comparison of Thin-Client Computing Architectures*. Location: Network Computing Laboratory, Columbia University. USA

Potts, J.R., Griffith Ph.D, T, Sharp, J.J., Allison, D. (2010). I/ITSEC Paper # 10357 Title: *Subject Matter Expert-Driven Behavior Modeling Within Simulation.* Location: I/ITSEC Archive. NTSA

Russell, S.J., Norvig, P. (2010). Book Title: *Artificial Intelligence: A Modern Approach (3nd Edition).* Location: Prentice Hall. New Jersey, USA.

Wooldridge, M., Jennings, N.R. (1995). Research Article: *Intelligent Agents; Theory and Practice*. The Knowledge Engineering Review, 10, pp 115-152. Location: Cambridge University Press. DOI:10.1017/50269888900008122