MODELING THE EFFECTIVENESS OF UNDERWATER SONAR

Jose A. Sepulveda
Luis Rabelo
Haluk Akin

Yanshen Zhu

Dept. of Industrial Engineering & Management Systems
University of Central Florida
Orlando, FL 32816, U.S.A.

School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816, U.S.A.

ABSTRACT

The purpose of this paper is to discuss an object-oriented simulation framework/model for the US Navy that simulates the behavior of ships in a combat environment. This model provides a framework which helps to improve threat recognition, undersea tactical awareness, countermeasure emissions, and counter-weapon fire control to enable surface ships to survive a salvo of torpedo attacks. A Navy ship gathers information from all the sensors it carries and then fuses the information to make a judgment. An Object-Oriented Programming (OOP) approach was adopted for designing and implementing our framework. With OOP, the framework can be easily modified and tested with sophisticated and classified algorithms that may be based on established physics principles. The model was developed by using multiple modeling approaches including hybrid, discrete and continuous systems and capturing complex phenomena. The model can be used to simulate different events in a combat environment.

1 INTRODUCTION

Today the US Navy faces many challenges in threat recognition. Foreign Navy platforms are now harder to find and possess a wider array of weapons (Vining 1999). With Navy ships operating closer to the littorals, more civilian objects such as yachts must be identified. These civilian objects add to the difficulties of quickly and precisely, detecting threats which may be a danger to the survival of the ship.

The Navy also faces the challenge of solving the claim they are harassing or killing whales with their sonar. Researchers and news organizations have reported that there is a connection between the sound produced by the sonar and that of the whales (Bressan 2004; Malakoff 2002; McCarthy 2002; Andre, Johansson, Delory, and van der Schaar 2005). For this reason it is also important for the Navy to evaluate and document the environmental effects of their sonar.

Much research has been conducted on threat recognition and its related decision making (Blake and Hummel 2004; Jeffers, Breed and Gallemore 2000; Totir, Radoi, and Quinquis 2005; Wilson 2003). The modeling algorithms developed by the researchers have helped to improve the effectiveness of the ship's sensor systems.

Today's budgetary, political, and above mentioned environmental concerns make it important to provide what the Navy calls "under way" training (Colitti-Howell Conway, and Dorsch 1997; Yardley, Thie, Schank, Galegher, Riposo, and Thie 2005). With simulation, we not only train the sailor in a virtual combat environment, but we also provide a means to evaluate the effectiveness of the different model algorithms and ship's equipment (van Doesburg, Heuvelink, and van den Broek 2005).

Navy surface ships tow an array of sensors, some passive and some active. These sensors are used to detect and classify all objects within a given detection range. The information from these sensors is then "fused" and a final decision to react or not to react is made. Many different sensor fusion technologies, such as using neural networks, have been developed (Boß, Diekmann, Jürgens, and Becker 2001), theory of evidence (Fabre, Appriou, and Briottet 2001), etc.

Today, more than ever, there is a need to use state-of-the-art technology to model and simulate the information gathering-data fusion-countermeasure firing process to evaluate the effectiveness of the system. With today's technology the Navy can decide which sensor or combat technologies to invest in and install on the current fleet.

This research team has designed and implemented a simulation model/framework at the University of Central Florida using Object-Oriented Programming. OOP was adopted because of its flexible benefits such as reusability, extensibility, and maintainability, and the fact that it has been proved to be efficient in modeling and simulation. (Berzins, Shing, Luqi, Saluto, and Williams 2000; Rida,

Boulmakoul, and Laurini 2003). The framework was implemented in Java using a simulation tool named Any-Logic™.

From the onset, we decided to implement the simulation model using an object-oriented design approach. Object-oriented design allows us to assign defined methods to do what they do to the different objects. For example, we may implement a detection method based in random numbers as follows: for each pair of sensors and threat we define a probability of detection at a given distance. We may even modify that probability as a function of the environmental conditions. But nonetheless, our detection algorithm essentially generates a random number and compares it to the stated value for the pair of sensors. If the generated random number is smaller than the specified threshold, it is concluded that this particular threat was detected by the sensor; otherwise, the threat was not detected. Object-oriented modeling permits later substitution of these crudely defined methods by sophisticated and classified algorithms that may be based in established principles of physics. Other advantages of object-oriented design are that it allows for user definition of the model's parameters and it can expand easily (for example, add different types of sensors and/or countermeasures as classes and objects in the model).

We also decided to implement our design in the Unified Modeling Language (UML), a tool designed for easy communications with third parties, while insisting on both a clarification of requirements and a glossary. As stated above, the simulation model itself was developed using AnyLogic by XJ Technologies, a professional simulation tool for accurately modeling complex hybrid, discrete and continuous systems and capturing complex phenomena by using multiple modeling approaches.

2 MODEL DETAILS

2.1 The Environment

The ocean is represented as a vast area without land masses. Distances are kept in nautical miles. Speeds are expressed in knots (1 knot = 1 nautical mile per hour). A ship's speed is modeled as changing according to an assumed distribution (see Figure 1).

2.2 Ship Model

The ship model defines the characteristics of a Navy surface ship and includes attributes such as number of anti-torpedo torpedoes (ATTs) carried, speed, location, bearing and patrol route. The ship patrols the ocean towing an array of sensors.
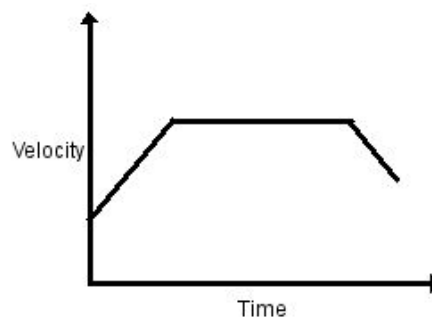

Figure 1: Ship Speed Distribution

Other vessels travel in the same general area. Their inter-arrival time, heading, speed, and type follow probability distributions. These vessels make noise and the modeled ship sensors may detect them if they are within their detection range.

2.3 Sensor Model

Sensors vary in their sensitivity (the ability to detect a threat) and their specificity (the ability to react to real threats only). For each sensor type, there exist curves (see Figure 2) that specify the probability of detection (in range and bearing) and of classification as a function of the threat type and distance.

We had no access to the data needed to fit these curves. Thus, we decided to create a set of Excel files which are easily modifiable by the user with the appropriate values. Table 1 illustrates this concept. According to the values provided in this table, the probability that a passive sonar will detect a civilian vessel type "1" at a distance of 30 [nm] is estimated at 0.80. The passive sonar will detect and correctly identify four out of every five (80%) civilian vessels of that type at this stated distance. Note that the probability increases to 19 out of every 20 (95%) when the distance decreases to 5 [nm]. Similar files are provided for other sensors and threats. In real life these probabilities change with the temperature, humidity, pressure, wind, and other factors. For the purpose of the model, the impact of these disturbances may be modeled with yet another hypothetical factor.

Table 1: Probability of Detection (by Type of Sensor) File

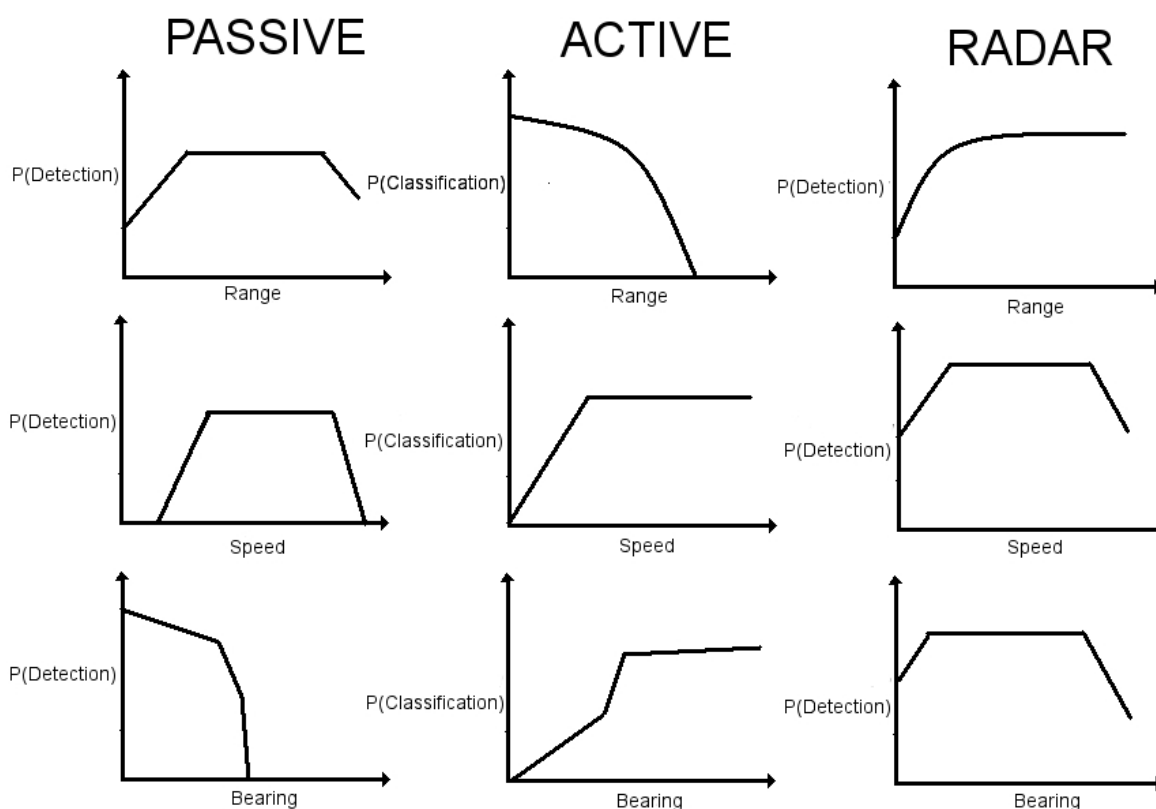| Distance	Civilian1	Civilian2	Civilian3	Civilian4
5	0.95	0.87	0.85	0.82
10	0.90	0.82	0.80	0.77
15	0.85	0.77	0.75	0.72
20	0.80	0.72	0.70	0.67
25	0.80	0.72	0.70	0.67
30	0.80	0.72	0.70	0.67

Figure 2: Probability of Detection (by Type of Sensor) Curves

2.4   Threat and Encounter Models

The threat and encounter models define the relative geometric relationships and likelihoods for threats encountering the modeled ship. A threat sub-model defines threat characteristics for motion and detection. There are civilian ships and marine animals among the threats in the detection range. Ships have a size, bearing, and speed (not necessarily constant). Marine animals (or natural phenomena that may be large enough so a sensor reacts to them) will occur at random and have a random path.

The threat model generates false and actual threats, and the range at which they occur. This is done through distributions whose parameters can be user-specified.

The encounter sub-model differentiates among encounter types within the detection range. Harmless threat encounters (blue icons are used with a numbered ID on top of the icon to display the threat type) are random ships or animals that cross the path of the modeled ship. Threat encounters are enemy ships (yellow icons) or torpedoes (red icons) launched from hostile ships. There is a small chance that the passive sonar will fail to detect a signal. As discussed in the previous example, the probability that, at 30 [nm], a type "1" civilian ship will be missed is 20%.

Ships have the ability to detect other vessels in their path and maneuver accordingly. We assume marine animals know how to avoid collisions with vessels.

2.5   ATT Effectiveness

ATT effectiveness is limited by the ability of the ship's sensors to recognize a threat attacking the ship and to react in time to prevent damage to the modeled ship. The detection and classification ranges are determined by the attack geometry of the threat weapons.

2.6   Sensor Sensitivity and Specificity

The goal of this project is to show how simulation can be used to assess the effectiveness of individual sensors and of the sensor system as a whole.

For any given sensor, the probability of detection and the probability of correct classification are rarely 1. There is always a chance that the sensor will not detect a signal and there is also a chance that, even if detected, the signal is misclassified.

There are two possible important errors: a harmless signal classified as a threat (risk of harming innocents and

wasting an ATT) and a signal from a threatening source classified as harmless (risk of being sunk). The simulation's objective is to model these occurrences and yield estimates of each sensor's (and the whole ship seen as a battery of sensors) effectiveness. To model this we have borrowed concepts of test sensitivity and specificity from the field of Public Health (Loong 2003). Figure 3 defines the different situations that may occur in the context of sonar and radar interaction with external signals and defines sensor sensitivity and specificity.

Sensor detects signal

- **True positives (TP):**
  - Torpedo is in the water
  - Sensor classifies the signal's source as a torpedo
- **False negatives (FN):**
  - Torpedo is in the water
  - Sensor classifies the signal's source is harmless
- **False positives (FP):**
  - No torpedo in the water
  - Sensor classifies the signal's source as a torpedo
- **True negatives (TN):**
  - No torpedo in the water
  - Sensor classifies the signal's source is harmless

- **Discovery Index = (TP + FN)/ (TP+FP+TN+FN)**
  - Chance of a torpedo attack
- **Sensor sensitivity = TP / (TP + FN)**
  - Proportion of signals from a threatening source correctly classified as threats
- **Sensor specificity = TN / (TN + FP)**
  - Proportion of signals from a harmless source correctly classified as harmless
- **Positive predictive value = TP / (TP + FP)**
  - Chance that a positive detection will be correct
- **Negative predictive value = TN / (TN + FN)**
  - Chance that a negative detection will be correct
- **Efficiency of the sensor = (TP + TN)/ (TP+FP+TN+FN)**
  - Fraction of signals correctly classified

Figure 3: Sensor Sensitivity and Specificity

Around the ship, any potential threat occurring within a sensor's detection area should be detected, classified, and localized. A passive sonar "listens" for signals and classifies the sound-emitting source as harmless or potentially threatening. This type of sensor detects most signals but may interpret a significant number of harmless signals as potential threats.

If a signal is detected, the sensor will classify it as threat ("positive" classification by the sensor) or harmless ("negative" classification by the sensor). If a signal classified as "positive" turns out to come from a threat source, the outcome for the situation is "TP" or "True positive" (positive because the sensor indicated a threat, true because

the classification turned out to be correct). By the same token, a "TN" or "true negative" is the outcome that occurs when the sensor classifies a signal as harmless and its source turns out to be so. In both of these situations the sensor behaved as desired.

There are two possible wrong conclusions by the sensor: a harmless signal classified as a threat (FP or false positives) and a signal from a threatening source classified as harmless (FN or false negatives).

Sensor sensitivity (the proportion of signals from a threatening source correctly classified as threats) and sensor specificity (the proportion of signals from a harmless source correctly classified as harmless) are important measures that would allow the Navy to decide whether a particular sensor device is worth deployment. The concepts may be extended to a set of multiple devices (passive sonar, active sonar, and radar) working as a coordinated set of devices (e.g., the decisions made in the ship's fire control room).

2.7 UML Diagrams

A UML diagram is a graphical representation of the elements of a system. Different diagrams allow you to view the system from multiple perspectives.

Figures 4 is an example of a UML class diagram, shown here to illustrate the concepts. Figure 4 displays some of the classes developed for the simulation, their attributes and the methods they use to communicate with other classes. All classes are drawn as a rectangle divided into three areas. The top area shows the class name, the middle area shows the class attributes (data associated to each object in this class), and the lower area shows the methods e.g., the coding for the exact sequence of steps the class uses to send messages to other classes or to respond to messages sent by other classes. For example, most classes inherit AquaticEntity from the parent class. A Ship is an AquaticEntity as it has (among other attributes) a speed and a heading and knows how to (e.g, has a method to do this) avoid a collision. An ArmedShip is a ship (has a speed, a heading and knows how to avoid a collision) but is different from a CivilianShip in that it carries torpedoes and anti-torpedoes. Similarly, a NavyShip inherits all the attributes and methods of an ArmedShip and adds a few other attributes and/or methods to their implementation: The NavyShip carries sensors that have methods to detect, classify and locate signals emitted by an EnemyShip, by marine animals, or by neutral vessels. It also has the ability to fuse the detected signals, realize that it is under attack and engage an attacking torpedo with an Anti-torpedo (ATT).
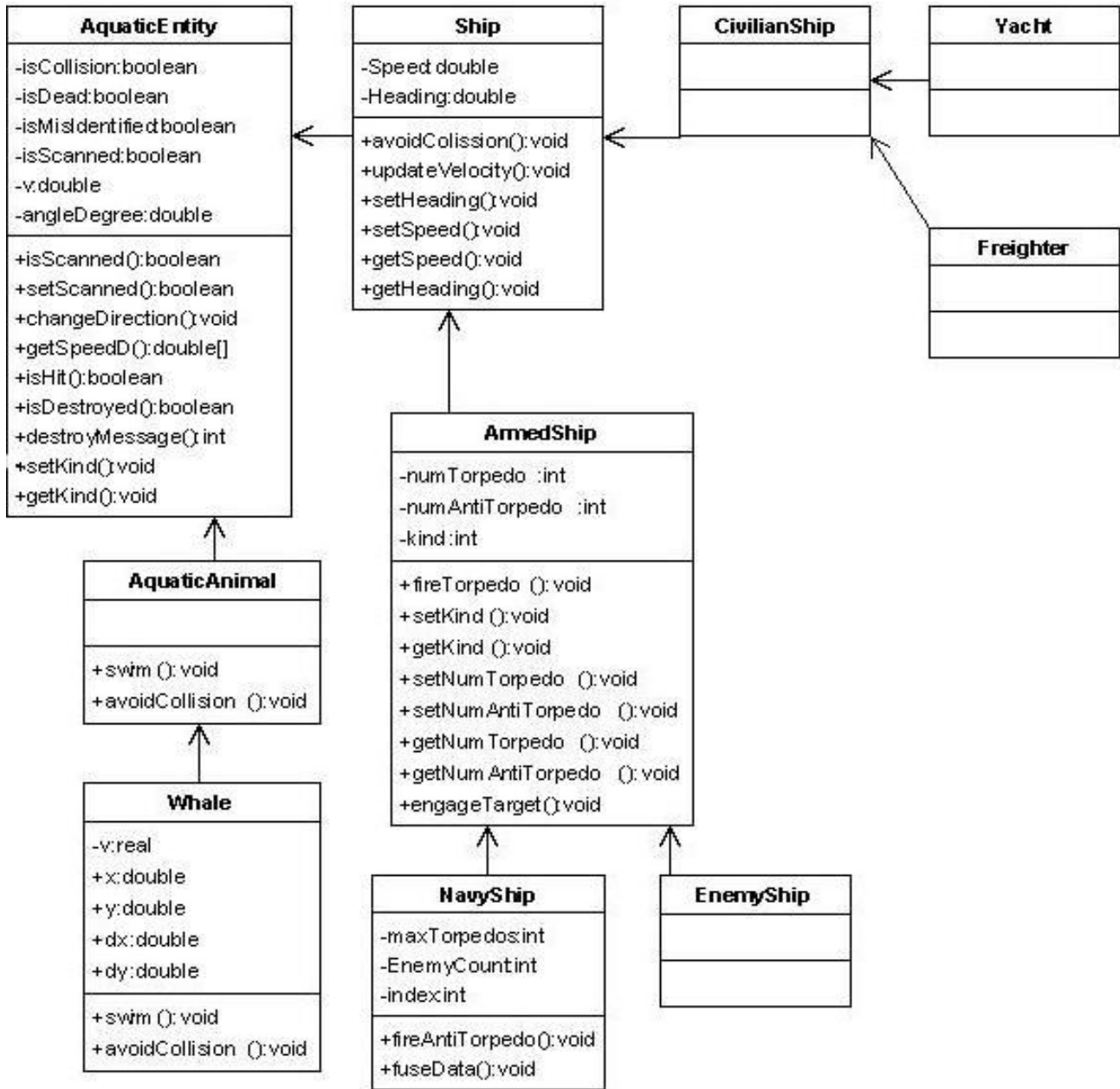
Figure 4: Class Diagram

2.8   Simulation Logic

The simulation model activates one ship and defines the available sensors and their corresponding sensitivity and specificity for each threat type. It also assigns to each ship a patrol route (heading and speed) and a complement of ATT. The ship patrols while scanning within its detection range.

The simulation then creates harmless objects and threat objects at random according to user-specified probability distributions. Most of the objects will be false threats, neutral ships or large marine animals or natural phenomena. Neutral ships have a heading and a speed, animals have random behavior.

Objects are disposed of when they exit the system boundaries. Information is written to an Excel file indicating whether the object traveled through a detection zone and, if so, if it was detected by the passive sensor.

Also occasionally a simulated enemy tries to get within range and launch a torpedo attack. The passive sonar should detect and correctly classify the threat. If the passive sonar fails to detect the enemy ship, there is still a chance that it will detect the incoming torpedo. In either case, the active sensor is engaged and should confirm the presence of a threat and activate the radar. If the radar confirms an underwater threat, the ship's fire control room engages the attacking torpedo when it gets within the response range.

The simulation will keep track (for each sensor and for the ship itself) of the false positives (no torpedo attack but sensor indicated the presence of a torpedo), true negatives (harmless signal and sensor classified it as harmless), true positives (attack detected by the sensor), and false negatives (attack classified as harmless by the sensor).

The simulation ends after all of the available ATTs are exhausted or all present ships are hit by a torpedo. The user can end the simulation at any time by using the "stop" button provided.

In addition to providing the sensitivity and specificity of each sensor, the same Measures of Performance are provided for the ship.

2.9   Simulation Parameters

As indicated, some of the simulation parameters are user definable. There is one Navy ship in the model. The region's patrolling area is about 1800 by 1200 nautical miles [nm]). There are no land masses in this area. The ship's average speed is 25 knots. Other vessel's average speed is modeled as a uniform distribution [3, 25] [knots]. The passive sonar's detection range radius is 30 [nm] and the response range radius is 10 [nm]. The torpedo (and ATT) speed are set at 45 [knots]. The ship starts with a complement of 20 ATTs.

Neutral vessels inter arrival time is modeled as a Poisson process with an arrival rate of three ships per hour. Enemy ships are modeled at the same rate while two animals are randomly generated per hour. The vessel origins is distributed as U [0, 360 degrees] (e.g., they can arrive to the region from any direction) and the vessels destination is modeled as U [origin + 45, origin + 270].

3   RESULTS

3.1   Output Screens

Figures 5 and 6 show typical simulation screens. In figure 6 the simulation ends when the ship was destroyed; Another cause for the simulation ending (not shown) occurs when the ship runs out of ATTs or when the user stops it (clicking on the Stop button in the screen's lower left corner).

The screen displays the main parameters for the run (they can be changed during the run) and a summary of the main observed results for each sensor and for the system as a whole. The number of undetected harmless and threat objects and the total number of ATTs left are also displayed and updated as they change throughout the simulation run.

In Figure 5 we see a routinely produced table at the end of the simulation. It summarizes the numbers reported in the simulation. We print to an Excel file each object's details and then summarize the results in a second file where all important statistics and observations are summarized.

4   CONCLUSIONS

A mathematical object-oriented simulation model that evaluates the effectiveness of alternative sensor configurations by providing measures of sensitivity (the ability to detect a threat) and specificity (the ability to react to real threats only) for each sensor and for the system (ship) as a whole was developed. This model used multiple modeling approaches to capture complex phenomena combining discrete and continuous systems and adding the possibility of implementing sensor fusion algorithms through agent-based modeling (Hoogendoom, Jonker, Schut, and Treur 2006).

This is a platform that integrates the simulation with legacy software (Excel spreadsheets or databases used to specify sensor and threat-related probabilities) as it executes a simulation run. The model also produces a number of Excel files that record the individual observations made for each object generated during the simulation runs.

5   FUTURE RESEARCH DIRECTIONS

A more advanced model would have land masses and navigational hazards which vessels must avoid (land masses may affect the sensor's capabilities). Therefore a geographical Information system (GIS) can be incorporated to the current simulation model to make the model more realistic.

Factors that impact critical ranges can be included in the model, for instance, the number of weapons, the presence of nearby ships, enemy and friendly countermeasures, the enemy's stealth ability, and the ship speed.
In the model we have mainly used Bayes' Theory. Adaptive Fuzzy Logic, Dempster Shafer algorithm, and a hybrid of Bayesian with Artificial Intelligence, Extended Kalman Filtering, and Neural Networks can be investigated as alternative algorithms to improve sensor fusion related calculations.
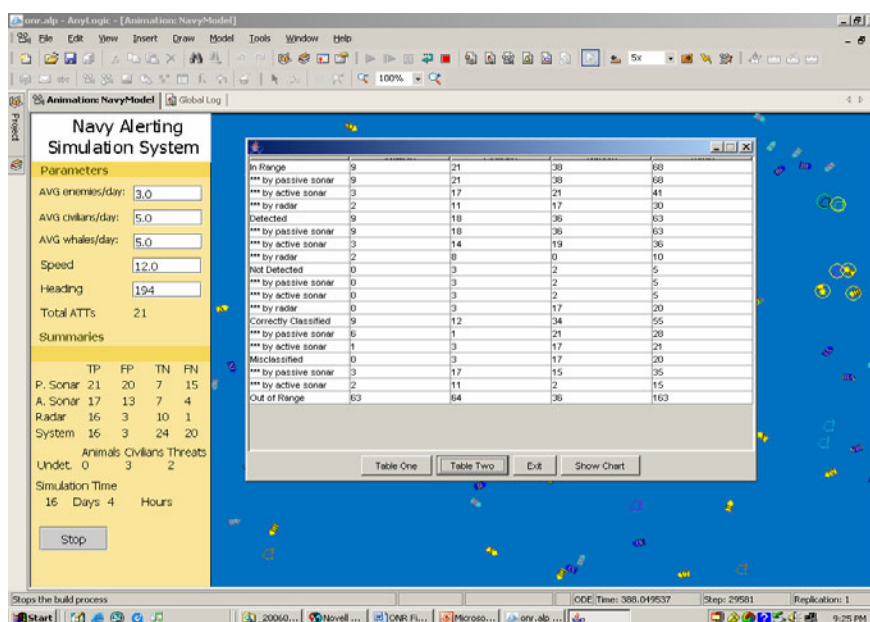
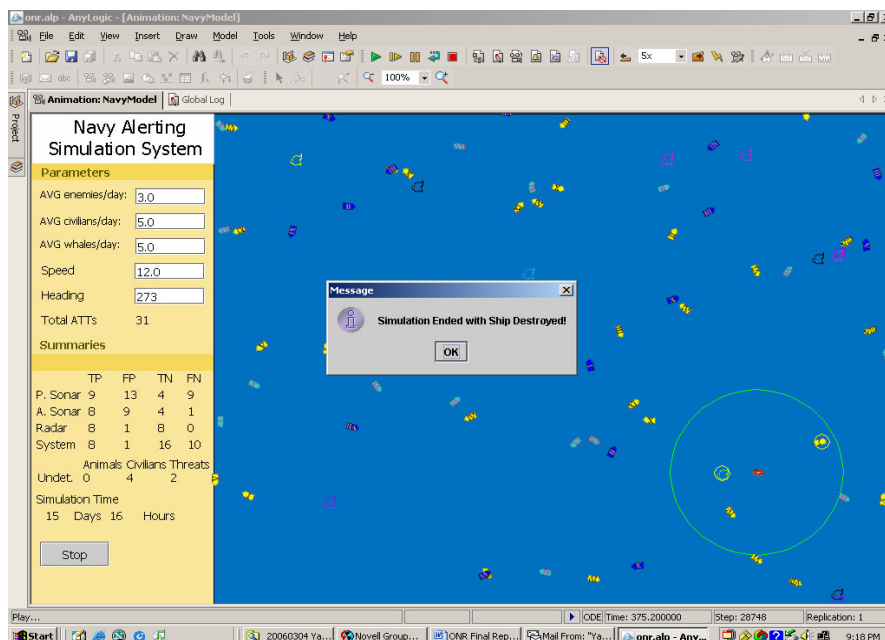Figure 5: Simulation Results Per Simulation Run

Figure 6: Simulation Ends with the Ship Destroyed

ACKNOWLEDGMENTS

REFERENCES

Andre, M., T. Johansson, E. Delory and M. van der Schaar. 2005. Cetacean biosonar and noise pollution. *Oceans 2005 – Europe.*

Berzins, V., M. Shing, Luqi, M. Saluto and J. Williams. 2000. Object-oriented modular architecture for ground combat simulation. *Proceedings of the 2000 Command*

and Control Research and Technology Symposium, *Naval Postgraduate School, Monterey, CA, June 26-28.*

Blake, W. D., and J. R. Hummel. 2004. Establishing human performance (Decision making) and natural environment consistency across integrated naval simulations. *72nd MORS Symposium.*

Bressan, R. J. 2004. Acoustic pollution in the puget sound and its effects on marine mammals executive summary. *Washington State University.*

Boß, T., V. Diekmann, R. Jürgens and W. Becker. 2001. Sensor fusion by neural networks using spatially represented information. *Biological Cybernetics.*

Colitti-Howell, M., J. Conway and S. Dorsch. 1997. Naval Surface Warfare Center Modeling and Simulation Master Plan. *Naval Surface Warfare Center.*

Fabre, S., A. Appriou, and X. Briottet. 2001. Sensor fusion integrating contextual information. *International Journal of Uncertainty, Vol 9, No. 3.*

Hoogendoom, M., C. M. Jonker, M. C. Schut, and J. Treur. 2006. Simulation, visualization, and validation of adaptive multi-agent organizations in naval applications. *Military Modeling and Simulation.*

Jeffers, R., B. Breed, and B. Gallemore. 2000. Passive range estimation and range rate detection, sensor array and multichannel signal processing workshop. *Proceedings of the 2000 IEEE.*

Loong, T. 2003 Understanding sensitivity and specificity with the right side of the brain. British Medical Journal. September 27, Vol 327.

Malakoff D. 2002. Suit ties whale deaths to research cruise, *Vol. 298 Science.*

McCarthy, M. 2002. Do we kill whales with sound. Discover, April.

Rida, M., A. Boulmakoul, and R. Laurini. 2003. Object-oriented approach and java-based distributed simulation for container terminal operational management. *In Proceedings of the 2003 International Conference on Information Systems and Engineering.*

Totir, F., A. Radoi, and A. Quinquis. 2005. Validation of Complex Naval Target Models using Superresolution Imagery Methods. *Oceans 2005– Europe.*

van Doesburg, W. A., A. Heuvelink, and E. L. van den Broek. 2005. TACOP: A cognitive agent for a naval training simulation environment. *AAMAS.*

Vining, P. 1999. What me worry? - The current state of surface ship torpedo defense. *Naval Institute Proceedings.*

Wilson, J.H. 2003. Detection and classification of submerged threats in very shallow water using the Quiet Interlude Processing System (QuIPS). *OCEANS 2003. Proceedings.*

Yardley, R. J., J. T. Harry, J. F. Schank, J. Galegher, and J. Riposo. 2005. Can under way training be reduced?

The Use of Simulation for Training in the U.S. Navy Surface Force.

AUTHOR BIOGRAPHIES

JOSE A. SEPULVEDA Ph.D., P.E., is an Associate Professor in IEMS at the University of Central Florida. He received an Ingeniero Civil Químico degree from the Universidad Santa María, Valparaíso, Chile, and MSIE, MPH, and Ph.D. (Industrial Engineering) degrees from the University of Pittsburgh. Dr. Sepúlveda's major areas of research interest are object-oriented simulation, risk analysis, catastrophe response, measuring and modeling training effectiveness, and applications of industrial engineering and simulation in health care. He has written two books andnumerous publications. His e-mail address is <sepulved@mail.ucf.edu>.

LUIS RABELO Ph.D., is an Associate Professor in Industrial Engineering and Management Systems (IEMS) at the University of Central Florida. He received dual degrees in Electrical and Mechanical Engineering from the Technological University of Panama and Master's degrees from the Florida Institute of Technology (Electrical Engineering, 1987) and the University of Missouri-Rolla (Engineering Management, 1988). He received a Ph.D. in Engineering Management from the University of Missouri-Rolla in 1990 where he also did Post-Doctoral work in Nuclear Engineering. He also holds dual MS degrees in Aerospace Systems Engineering & Management from the Massachusetts Institute of Technology (2001). His experience includes Ohio University, BF Goodrich Aerospace, Honeywell Laboratories, NIST, NASA, and MIT. His e-mail address is <lrabelo@mail.ucf.edu>.

YANSHEN ZHU is a Ph.D student at the School of Electrical Engineering and Computer Science of University of Central Florida where he received his M.S. degree in Computer Engineering in 2003. He received his B.S. degree from Wuhan University, China in 1998. His e-mail address is <yanshen_zhu@hotmail.com>.

HALUK AKIN is studying towards his Ph.D degree at the Industrial Engineering and Management Systems Department at the University of Central Florida where he received his M.S. degree in 2004. He received his B.S. degree from Yildiz Technical University, Istanbul in 2002. His e-mail address is <halukakin@gmail.com> .