Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

DYNAMIC PATH-PLANNING FOR SEARCH AND DESTROY MISSIONS – THE BAY OF BISCAY SCENARIO

Subhashini Ganapathy
Raymond R. Hill

Department of Biomedical, Industrial and Human Factors Engineering
Wright State University
207, Russ Engineering Center
Dayton, OH 45324, U.S.A.

ABSTRACT

Among the many modeling methods used for military applications, simulation modeling is one of the most popular as it offers flexibility and an ability to perform "what-if" analysis. In this paper, we discuss search and destroy missions in the context of the World War II Bay of Biscay U-boat scenario. We present a simulation architecture that supports integration of human reasoning with simulation-based optimization methods.

1 INTRODUCTION

Simulation models are used extensively for studying military applications. The primary goal of a military simulation is to provide a high fidelity representation of the combat conditions. Among the many modeling methods used for military applications, simulation modeling is one of the most popular as it offers flexibility and an ability to perform "what-if" analysis (Battilega and Grange 1978). In the Search and Destroy (SAD) missions, a troop would search for the enemy and when found would capture or destroy the enemy force. Typically, in such cases, there is a threat to the search team by the enemy targets, directly or indirectly. In a direct threat there are attacks without an intervening agency or resource acting on behalf of the threat.

In a SAD scenario the information about the objects of interest (targets) is very important and is dynamic in nature. There could be two situations, (a) searching for an elusive target, and (b) if a target is located and is subsequently lost. In the latter case, it behooves us to start the search in the area the target was last located. The knowledge about the target location (search area) temporally decays and this knowledge may no longer be relevant with change in time.

The dynamics and uncertainty associated with this problem scenario makes it very difficult to apply purely analytic approaches. Human reasoning integrated with heuristic optimization methods offer a potentially attractive alternative.

In this article, we present the simulation architecture in the context of Bay of Biscay scenario. We present the decision algorithm that the simulation architecture uses in planning the path over the search area. Our model implements the computational infrastructure to support reusable software components. The architecture represents a dynamic path planning in the context of the Bay of Biscay scenario involving interactions associated with search for targets with minimum resources and maximum efficiency. Simulation models give the flexibility of expanding and compressing time (real world time) and, thus, allow extensive investigation of the system during any period of time. It accounts for the dynamics and uncertainty of the system.

2 PROBLEM DOMAIN

This study focuses on the U-boat hunting problem during World War II in the Bay of Biscay area. The Bay of Biscay is an inlet of the Atlantic Ocean in southeastern Europe, bounded by France and Spain. The German U-boats operated from captured ports in occupied France, crossing the Bay of Biscay to gain access to the North Atlantic. In the Atlantic, the U-boat wolf packs would look for and attack allied convoys. The Allied convoys provided critical logistical support to the war effort. The convoys would ship basic necessities and war material to Great Britain. As counter measures, the Allied force sent out airplanes to search and destroy the U-boats. According to McCue (1990), one of the biggest challenges faced by the Allies was the Axis U-boat threat.

U-boats most imperiled North Atlantic shipping in 1942 and 1943. The Allied forces sent aircraft to search the Bay of Biscay area for U-boats that were in transit between their ports in occupied France and the areas in North Atlantic. This search effort was more of an "offensive" campaign than the "defensive" task of protecting convoys (Waddington 1973).

The problem domain can be described by the entities present in the system, their states and events as shown in Figure 1. The U-boats can be present in one of the three states: surfaced, submerged or sunk. The aircraft remain either on the base to be deployed or on the bay searching for the U-boats. The three major functional components of the system are:

1. The place of origin of the U-boats (French Ports & German Shipyard) and North Atlantic
2. The Bay of Biscay
3. The place of origin of the aircraft (airfields in England)

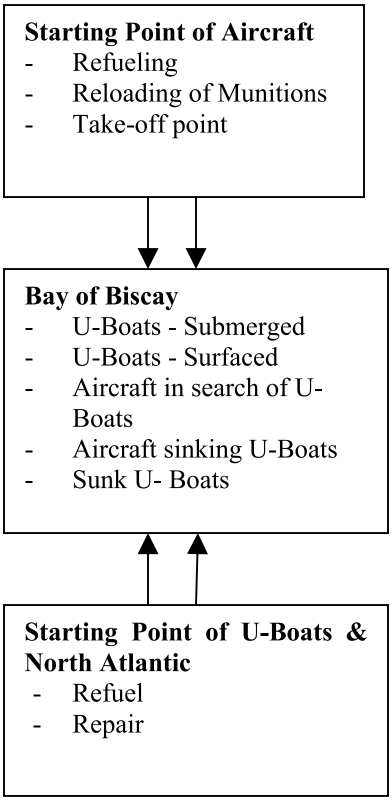

Figure 1: List of Entities, Their States and Events

There are two events that occur in the ports of occupied France. The U-boats returning to France undergo repair and service. The aircraft return to the base for refueling and reloading of munitions. Sinking of U-boats in the Bay depends on the number of U-boats surfaced and submerged, and the Allied search effort.

The U-boat war presents an interesting study, particularly when examining the dynamic decision processes in either side. The technologies used by both sides kept changing and updating with new strategies and countermeasures. The British introduced Anti-Surface Vessel Mark II airborne radars to facilitate night time search. The Germans as countermeasure changed their strategies for transit by lessening detectability and vulnerability. The Allies tried to match their search effort balance with that of the Germans strategy of day and night travel.

McCue (1990) developed a mathematical model to study the behavior of the system that is known as the U-boat Circulation Model. The U-boat circulation model enables evaluation of performance of different Bay measures and countermeasures, using merchant ships saved as a measure of performance.

3 RESEARCH APPROACH

There has been a lot of research on developing algorithms for finding shortest paths in a particular topology or map. Many of the previous models have failed to include the dynamics and uncertainty of the planning model. Stentz (1994) presents planning trajectories for mobile robots but the model assumption that the environment is predetermined is inaccurate. Canny and Lin (1990) discuss a robot path planning algorithm that constructs a global skeleton of free-space.

A discrete search space consists of a network of arcs interconnected with or without repetition of the nodes. Some of the well-known search algorithms for state space search include A*, Dijkstra and Ant-Search (Cormen, Leiserson, and Rivest 2000; Gallo and Pallotino 1988; Nilsson 1982;).

Dijkstra's algorithm has been the widely used algorithm because it is simple and has less computation time. Some of the applications include vehicle routing, emergency routing and response system.

These algorithms aid in finding a solution that takes the form of a sequence of actions leading in a path from the start state to a goal state. One of the major disadvantages of these methods is that they are deterministic in nature. These algorithms fail to account for the temporal change in the environment. One of the limiting factors for destroying the U-boats was the amount of munitions carried by the airplane. The aircraft could carry only one bomb at a time and once they dropped their bomb they needed to fly to the base and rearm. The problem becomes very interesting for the following scenario: If an airplane sights a U-boat on its way back to the base, what is the flight path of the airplane after rearming? This can result in two alternatives:

1. The airplane can return back to the sighted location and search for the U-boat
2. The airplane can go and follow its predefined search pattern.

Given a particular region the aircraft can travel along any path. The choice between the alternatives depends on the cost effectiveness of the alternatives. The cost function depends on the probability of finding the U-boat at the sighted location. Good search heuristics can result in improving the quality of the solution in terms of the resources spent. Richardson (1980) discusses the United States Coast Guard Computer Assisted Search Planning System (CASP) development work that investigates the search plan over a rectangular region.

The process of planning involves six steps. The first step is to define the objective and second is the correct problem formulation. The third step involves generation of alternatives and then comes evaluation of the alternatives. The next step is the choice of alternatives. The final step could be a feedback loop to give input to the simulation (Hozaki and Iida 2001).

The decision process for selecting the path is as shown in Figure 2. The base for the decision process is a Dijkstra's algorithm, that also checks for optimal resource allocation based on the aircraft location, the resources present for the aircraft and whether it is armed or not. Dijkstra's algorithm can be used to find the shortest path on a weighted, directed graph with a single-source node. The Bay area is divided in to equal segments of 100NM and each intersection of the square is considered a node. Each node is associated with a weight based on the distance between the nodes, the amount of fuel required to cover the distance, and how recently an aircraft has visited the node. The route of the aircraft is determined by first checking to see if the aircraft has ammunition. If the aircraft needs to be rearmed then it is routed back to the base, otherwise the algorithm calculates the next node based on the weights. Once the next location is calculated this information is presented to the user, and they can change the allocation based on the heuristics.

The search area covered can be expressed as the operational sweep width of the airplane. Operational sweep rate measures submarine sightings in square miles per hour.

4 COMPUTATIONAL ARCHITECTURE

High-fidelity computer models and simulations are very useful in design as well as ongoing systems management of many complex, real-world, dynamic systems (Narayanan et. al. 1998). The system developed for this study is a discrete event simulation.

4.1 Simulation Architecture

The simulation architecture is based on an object-oriented architecture and supports interactive simulations (Narayanan et. al. 1997). The different modules of the system are (a) Basic Simulation Module, (b) Allied Resources Module, (c) Interface Module, and (d) German U-boats Module.

Interactions between different components are represented using UML as shown in Figure 3. UML is a language for specifying, constructing, visualizing, and documenting the artifacts of a software-intensive system (Muller 1997).

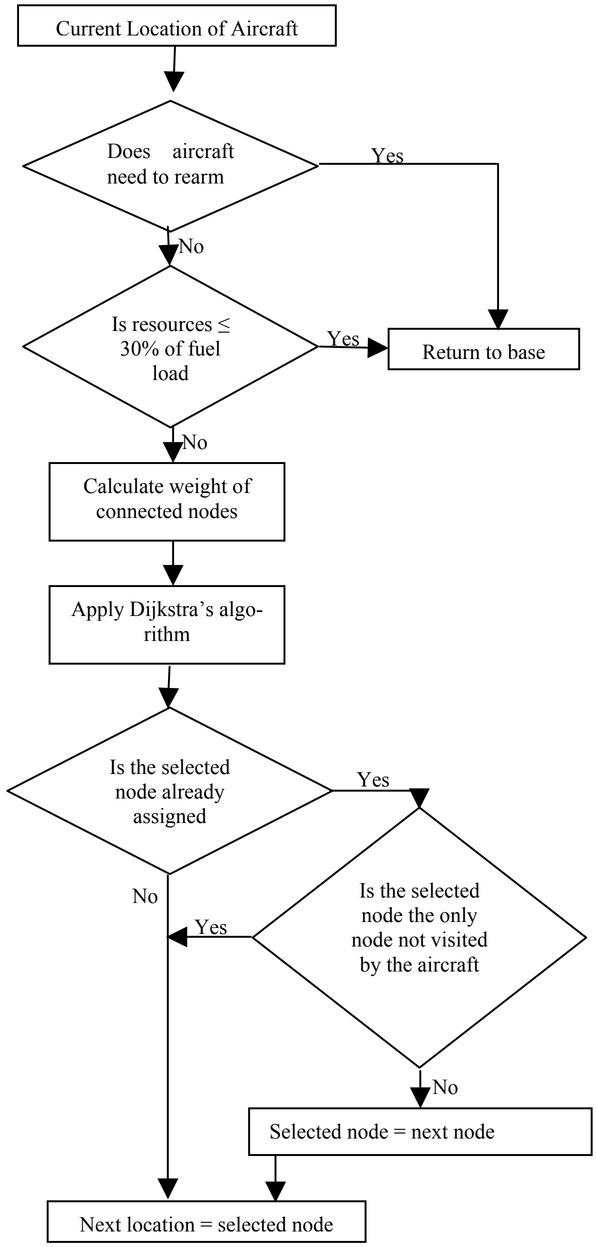

Figure 2: Decision Process for Selection of Waypoints

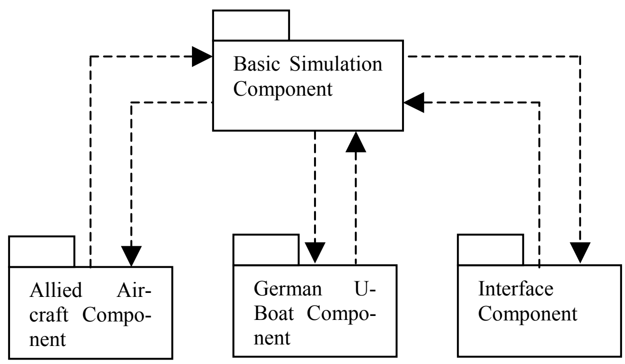

Figure 3: Dependency Diagram of Simulation Architecture

Within the simulation, the entities are connected in a multi-threaded architecture. Hence, each individual aircraft or U-boat can be controlled and can describe different behavior. The architecture is built using the Java 2 environment and can operate across different platforms. The basic simulation module is responsible for the scheduling of the events and coordinating the multi-threaded architecture. It runs a standard simulation clock that tracks the time of the simulated system. The interface is developed using the Java swing package.

Choosing the distributions for the occurrence of events is very critical in a simulation. There are two types of distributions used for this architecture: uniform and exponential. In uniform distribution, the minimum and maximum are provided and all values within the range are equally likely to occur. The set of random numbers can be replicated; hence, these distributions generate pseudorandom numbers. The simulation design data is based on historical data and military expert opinion.

4.2 Features of the Model

The simulation interface is shown in Figure 4 . The time period for modeling the simulation was chosen as April 1943 – September 1943. During this timeframe the technologies used by both the sides were fairly steady.

4.2.1 Aircraft Module

The aircraft leave from a single base at Great Britain and search the base area based on the modified Djikstra's algorithm. They standoff 100 NM from the coast of France to avoid enemy air patrols and escorts (Champagne and Hill 2003). They can sight a U-boat only when the U-boat travels on surface. The nodes are assigned based on the dynamic path-planning algorithm.

The aircraft search around the middle of the bay area as shown in Figure 4. They do not go too close to the French ports as the U-boats had support convoys. The aircraft are limited by the munitions they can carry. They are also constrained by the search space and the capacity of the aircraft. Some of the assumptions made in this model are (a) the aircraft return to the base once the resources get depleted to 30% of their total capacity, (b) U-boats and the airplane move at a constant speed and do not accelerate, and (c) U-boats are always sighted when they are directly below a search beam of the airplane.

Some of the major decisions based on heuristics are: if a aircraft finds a U-boat on its return to the base for refuel then it would mark that place and let the other aircraft know about the specific location; if there are aircraft closer to that region then they would go and bomb that U-boat.

One of the limiting factors for destroying the U-boats was the amount of munitions carried by the aircraft. The

Figure 4: Simulation Interface

aircraft could carry only one bomb at a time and once they dropped their bomb, they needed to fly to the base and rearm.

4.2.2 U-boats Module

The U-boats travel from five different ports and then cross the Bay of Biscay at different points in the sea. At the beginning of the simulation the U-boats are assigned randomly to any port based on a uniform distribution.

The simulation introduces anomalies in the system in terms of false targets, whose detection characteristics are identical to that of the target. A false target could represent a tide or water turbulence. Such false targets are called false positive. These false targets have the same detection function as the real target. The changes in the simulation module can be viewed on the interface as animations.

5 DISCUSSION

We have presented here an initial attempt to integrate human reasoning and decision making into a complex model. The computational architecture is generic and can support various studies related to heuristics optimization. In evaluating the effectiveness of the search it is important to properly formulate the effort. One of the most common measures in a search and destroy scenarios is to note whether a target has been found or not. If the target is not found then this measure does not give any clue regarding the thoroughness and efficiency of the search nor does it provide guidance on when to stop the search. Hence it is important to choose the right measure in assessing the system.

The different studies that are in progress are (a) developing software patterns that model prototypical scenarios and implement them as a library of classes using an object-oriented programming language, (b) formulation and implementation of heuristic optimization methods, and (c) exploring the effectiveness of human integrated decision-making system with respect to an automated system.

ACKNOWLEDGMENTS

The research was sponsored by Defense Modeling and Simulation Office through the Air Force Institute of Technology.

REFERENCES

Battilega, J. A., and Grange, J. K. 1978. *The Military Applications of Modeling*. Air Force Institute of Technology Press: Wright-Patterson Air Force Base, Ohio.

Canny, J., and Lin., M. C. 1990. An Opportunistic Global Path Planner. In: *Proc. IEEE International Conference of Robotics Automation*. 39-48.

Champagne, L. E., and Hill. R. R. 2003. Multi-Agent Simulation Analysis: Bay of Biscay Case Study. In *Proceedings of SimTecT 2003 Conference.* Adelaide, Australia.

Cormen, T. H., Leiserson, C. E., and Rivest, R. L. 2000. *Introduction to Algorithms*. MIT Press: Cambridge, MA.

Gallo, G. and Pallotino, S. 1988. Shortest Path Algorithms. *Annals of Operations Research*. (13): 3-79.

Hohzaki, R., and Iida, K. 2001. The Process of Search Planning: Current Approaches and Continuing Problems. *European journal of operational research,* 133: 120 – 129.

McCue, B. 1990. *U-boats in the Bay of Biscay: an Essay in Operations Research*. National Defense University Press: Washington D.C.

Muller, P. 1997. *Instant UML*. England: Wrox Press.

Narayanan, S., Scheider, N. L., Patel, C., Reddy, N., Carrcio, T. M., and DiPasquale, J. 1997. An Object-Based Architecture for Developing Interactive Simulations using Java. *Simulation*, 69 (3): 153–171.

Narayanan, S., Bodner, D. A., Sreekanth, U., Govindaraj, T., McGinnis, L. F., and Mitchell, C. M. 1998. Research in Object-Oriented Manufacturing Simulations: An Assessment of the State of the Art. *IIE Transactions,* 30 (9): 795–810.

Nilsson, N. J. 1982. *Artificial Intelligence*. Springer-Verlag, Berlin Heidelberg: New York

Richardson, H.R., and Discenza, J.H. 1980. The United States Coast Guard Computer-Assisted Search Planning System (CASP). *Naval Research Logistics Quarterly*, Vol. 27, 659-680.

Stentz., A. 1994. Optimal and Efficient Path Planning for Partially-Known Environments. In IEEE Int. Conf. Robot. & Autom., 3310-3317.

Waddington, C. H. 1973. *O.R. in World War 2: Operations Research against the U-Boat.* Paul Elek (Scientific Bools) Ltd.: London, England.

AUTHOR BIOGRAPHIES

SUBHASHINI GANAPATHY is a Research Assistant at Human Computer Interaction Lab, Wright State University. She is currently pursuing her Ph.D. in Humans in Complex Systems. Her research interests include predictive analysis in model-based information technology systems, design optimization, and simulation and modeling. Her email and web addresses are <sganapat@cs.wright.edu> and <www.wright.edu/~ganapathy.2>.

RAYMOND R. HILL is an Associate Professor of Industrial and Human Factors Engineering with Wright State University. He has a Ph.D. from the Ohio State University and has research interests in heuristic analysis, applied optimization and simulation modeling. His email and web addresses are <ray.hill@wright.edu> and <www.cs.wright.edu/~rhill>.