An Agent Based Proposal for Modeling Defense Behavior in Amphibian Operation Simulation System

GUSTAVO HENRIQUE SOARES DE OLIVEIRA LYRIO
ROBERTO DE BEAUCLAIR SEIXAS

PUC-Rio–Pontifical Catholic University
TECGRAF–Computer Graphics Technology Group
Rua Marquês de São Vicente 225, 22453-900 Rio de Janeiro, RJ, Brazil
glyrio,rbs@tecgraf.puc-rio.br

IMPA–Institute of Pure and Applied Mathematics
Estrada Dona Castorina 110, 22460-320 Rio de Janeiro, RJ, Brazil
glyrio,rbs@impa.br

Abstract. Simulation Systems today, are a valuable tool in the process of military training. These systems often are divided in two different sides, attack and defense. The force been trained provide "players" for both sides, training both attack and defense tactics. In case of Marine Corps, that isn't nice. Defense isn't a Marine Corps attribution. Marines are trained to conquer and secure an area of few kilometers over a beach to allow a secure Army approach. So, the group of marines that takes responsibility over defense has their training lost. To avoid that problem we present in this work an artificial intelligence solution, based on agents' behavior, modeled using finite states machines. With the presented behavior agents will be able do figure as the defense force allowing all marines to figure as attack.

Keywords: Military Modeling and Simulation, Agent-based Combat Modeling.

1 INTRODUCTION

We have been working with Brazilian Marine Corps in the development and maintenance of combat strategic didactical systems in warfare simulations along many years. The goal of these systems is to simulate combat strategies and lines of action used by Brazilian marine officer students allowing the validation and evaluations of these strategies. The systems are mainly double sided action [5]. It means that both attack and defense forces are controlled by the marine officers. As the defense is not usually a task attributed to the Marine Corps in an amphibious operations, the group of students that receives these task may have their learning harmed. To avoid such harm we have introduced artificial intelligence in our systems using the agents' concept to model military defense units. A unit can be any fraction of a military force (platoons, companies, etc.)

To create an artificial intelligence model of an unit we should look upon the way it thinks and identify its mainly characteristics [4]. The first characteristic that appears when we look upon a military defense unit way of thinking is that they don't act by their on will. Military units, attack and defense, obey orders from a higher command and follow combat strategies and doctrines. Their behavior is well know and described in marine training documents [3]. That characteristic leads us to build our agents model using a predefined behavior that bases the units behavior on the techniques described in marine documentation and on the knowledge of a instructor officer.

This paper presents a new approach to quickly model agents with predefined behavior using finite state machines to simulate a defense force controlled by a higher command.

Why a higher command ? Although the decentralization is well know as the most adopted philosophy of command and control, some decisions still have to be taken by a higher command, which knows the situation, advantages and weakness of all units on the battlefield and is also able to coordinate their efforts providing a best defense. We will now start explaining the modeling of that command, followed by the modeling techniques, and the doutrine and behavior of military units.

2 COMMAND MODELING

Using agents to model military defense units brings the necessity of some kind of centralized command. Units can't stand on terrain an act for themselves. Otherwise it will turn into a model of decentralized focus of resistance and not a military defense model. It's necessary some kind of centralization to coordinate these units. In a real combat scenario this paper will be played by a command post that receive information from the units, analyze the scenario and take global decisions, such as to request fire support missions, for example [2].

In this model, agents will stand on their positions until some threat is spotted. Then the agent that has spotted the threat put the spotted position on a target list (simulating the message "enemy spotted" sent to command post). After all agents turn, the command post analyzes the target list verifying whether a target can be hit by artillery, which one will demand aerial fire and which does not represent an immediate threat. Then the command post chooses the better positioned elements to provide fire support, and requests the missions (Figure 1).

3 AGENTS MODELING TECHNIQUES

> "An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors." [4]

Starting from this definition, the goal is to identify what is relevant to be perceived and which actions should be taken by the agent. Two models are needed here: sensors and effectors.

Our sensors were modeled by questions that an agent needs to answer when inserted in the environment. The answers for these questions are always true or false. We grouped some of these questions and their answers to represent situations where an agent can be involved with. We call these situations states. The more complex is the model, the higher is the number of sensors and states.

Now that we have found a way to model sensors, we need to model effectors. We will do that, defining actions that the agent will take when in a state. We should also, identify which events in the environment may change our agent's state. We call these events transitions.

Let's model a sentinel on a tower, for example. The tower will have an alarm switch that the sentinel will turn on if some threat approaches. The guy has two sensors. One defines if he can see some enemy coming while the other defines if the alarm is on. So, we got 2 states: watching and sounding alarm state. When watching, the sentinel takes an action of look upon horizon to verify if any enemy is spotted. If no one is spotted, the sentinel remains in watching state, but if an enemy is spotted a transition is made to sounding alarm state. The sentinel then will take the action to reach and sound the alarm and return to watching state.

It's true that a sentinel modeled this way is too far from reality. Among many other things, a sentinel needs to rest, he can't remain watching forever. Well, you can always improve the model by adding more states and transitions. Let's suppose now that our sentinel has an attribute called energy that goes from 100% (fully active) to 0% (extremely tired). When in watching state the sentinel's energy is decreased by 1% each 7 minutes. If it reaches 20% the sentinel will change to a new state called sleeping. When in that state, the sentinel will recover 3% of his energy each 7 minutes and change to watching state when the energy reaches 100% again. Again we can add many more states to make our sentinel more real, but the goal here is only to exemplify how the idea works.

As we can see in Figures 2 and 3, this approach agrees perfectly with the concept of finite state machines.

> "A finite state machine (FSM) consists in a set of states (including an initial state), a set of inputs, a set of outputs, and a state transition function. The state transition function takes the input and the current state and returns a single new state and a set of outputs. Since there is only one possible new state, FSMs are used to encode deterministic behavior." [1]

Our FSM initial state will be the initial agent state. In our sentinel example, the watching state. The inputs will be the sensors and the outputs will be these sensors updated. Finally, transitions are conditions that have to be fullfiled to allow state transition function to modify the current state, like some enemy coming or alarm turned on.

Summarizing, we are using finite state machines to model agent's behavior. For that we group one or more sensors to make the states. At each state the agent will do one or more actions over himself or over the environment. These actions are our effectors. Some events will make the agent switch from one state to another. These are the state machine transitions.

4 DOCTRINE AND BEHAVIOR OF DEFENSE MILITARY UNITS

Our defense unit agent's model is based on attrition warfare. We have chosen attrition warfare because this is the most common defense doctrine. The attrition warfare can be summarized as (citar fonte)

- Cumulative destruction of enemy through application of superior firepower;
- Minimize the enemy strength;
- Centralized control;
- Terrain captured (defended in our case) is the primary battle metric.

Using that summary as goals, and with the help of a defense instructor from Brazilian Marine Corps, we divided agent's behavior in two different kinds of defense behavior: position defense behavior and enemy delay behavior.

Since we are modeling defense units, there are many common sensors in both behaviors. A unit having a defense behavior needs to be able to see when the enemy arrives, so our first sensor will be "Can I see some enemy?". From that we can define an effector that is the action took by the unit when no enemy is spotted, to look for threats. When an enemy is spotted the unit will need to know if it's a new enemy or if it has already been spotted before. That will avoid the unit from keep requesting fire over the same target. This demands our next sensor "It's a new target?". It also demands a new effector called request fire. Due to the huge reach of some artillery armaments, the unit will probably receive artillery fire from the enemy even before the

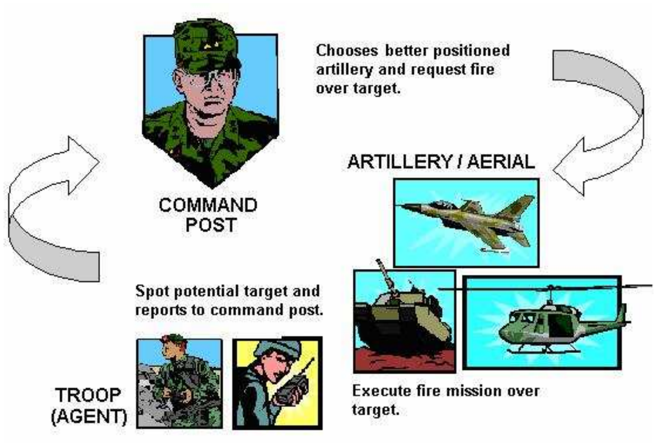

Chooses better positioned artillery and request fire over target.
COMMAND POST
ARTILLERY / AERIAL
Spot potential target and reports to command post.
TROOP (AGENT)
Execute fire mission over target.

Figure 1: Schema showing command post activity.

Agent Type	Sensors	Effectors	Transitions
Sentinel	Is some threat ?	Look for enemies	Enemy spotted
	Is the alarm on ?	Reach and sound alarm	Alarm turned on
Sentinel Improved	Is someone coming ?	Look for enemies	Enemy spotted
	Is the alarm on ?	Reach and sound alarm	Alarm turned on
	Energy is less than 20%	Sleep	Energy fully restored (100%)

Table 1: Table showing sensors, effectors and transitions for both agent model examples.

enemy is spotted. Therefore, it will need to calculate the enemy's possible position (a new effector) to request fire back over the enemy. So, a new sensor is "Am I under artillery fire?". The unit also will need to know when attacked by direct fire, then another two common sensors are "Am I under direct fire?", and "Can I engage the enemy?". Thus, our last effector is to engage the enemy. The last common sensor is related to the number of men in each unit. It will be necessary to determine when the unit loses half of its men. That is a critical value that needs a new sensor "Have I lost half of my men?". This sensor will be a particular one in our model because the effector that it demands depends on the behavior chosen.

Once we have defined the common sensors and effectors, it is necessary to remember that we have two different behaviors to model: the position defense and the enemy delay.

When defending a position a unit can't retreat and when the causalities reach half of the unit troop it will request reinforcements and wait until they arrive. Our particular new effector and sensor are, respectively, "Have the reinforcements arrived?" and "reinforcements arrived".

However, if the goal is to delay the enemy when the causalities reach 50%, the unit should retreat to another position on terrain. This provides a new particular effector, to retreat. The unit will need to know when the new position is reached. This is our new sensor "Have I reached my new position?".

Now that we have defined our sensors and effectors for both behaviors, let's group then in states, actions and transitions to build our finite state machines.

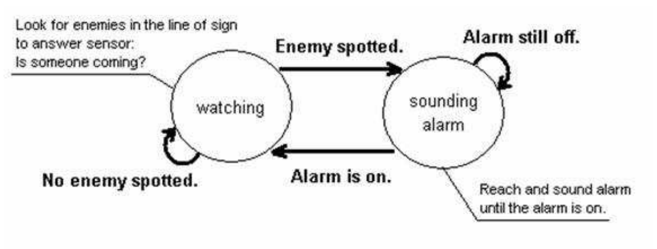
4.1 Defense Position Finite State Machine

4.1.1 Idle state:

This is the default combat element state. It will group all the sensors with negative answers. No threat was spotted, no new target, no enemy artillery fire taken, unit is not been attacked, 50% casualties not reached, no reinforcements are waited. When in that state the unit holds its position and have only one effector: look for any threat in its vision sight (Can I see some enemy?). The transitions here are done when some of the sensors change its answer.

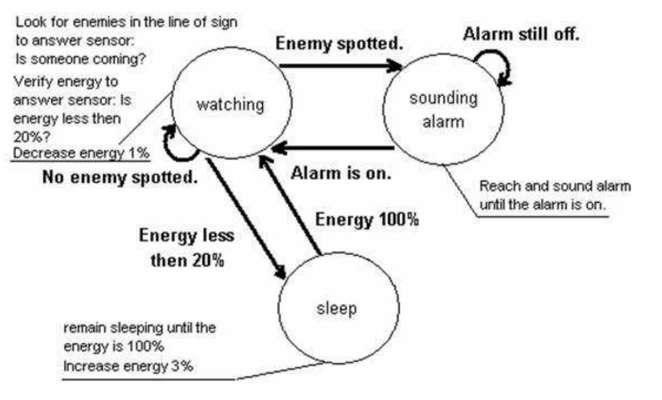
4.1.2 Threat spotted:

This state groups vision sight sensor (Can I see some enemy?) and new target sensor (It's a new target?), both with affirmative answers. The effector here is to request fire over target. When an enemy is inside the unit's vision sight, the unit will verify either if the enemy position isn't listed or if it is listed as an already fired target. If the position is not listed yet, the unit will insert the threat position on the target list as an unfired target. If the threat position is already listed as fired target, the unit will change its status to unfired, requesting a new fire mission over the threat. However, if the threat position is listed as an unfired target the unit ignores the enemy, which means the command post knows the enemy but wasn't able to request a mission to that target yet. When the command post requests a fire mission on a target it will change the value to fired target. It's



Look for enemies in the line of sign to answer sensor: Is someone coming?
Enemy spotted.
Alarm still off.
watching
sounding alarm
No enemy spotted.
Alarm is on.
Reach and sound alarm until the alarm is on.

Figure 2: Automatous diagram for the sentinel agent model.



Look for enemies in the line of sign to answer sensor: Is someone coming?
Verify energy to answer sensor: Is energy less then 20%?
Decrease energy 1%
Enemy spotted.
Alarm still off.
watching
sounding alarm
No enemy spotted.
Alarm is on.
Reach and sound alarm until the alarm is on.
Energy 100%
Energy less then 20%
sleep
remain sleeping until the energy is 100%
Increase energy 3%

Figure 3: Automatous diagram for the improved sentinel agent model.

necessary to avoid multiple fire requests to the same target. After inserting the threat on the target list or ignoring it, the answer to both sensors will change to false and a transition will be made to the idle state. That keeps the unit checking when the requested fire missions are accomplished.

4.1.3 Under enemy artillery fire:

The only sensor in that state is the artillery fire received sensor (Am I under artillery fire?). When under enemy artillery fire, the unit will call its effectors: calculate the direction and position where the fires came from and request fire over enemy.

4.1.4 Attacked:

Attacked state has two sensors with affirmative answers: being attacked sensor (Am I under direct fire?), engagement possible sensor (Can I engage the enemy?), and the causalities sensor (Have I lost half of my men?) with negative answer. An agent will enter this state when receive direct attacks from enemies. It calls the engage enemy effector to strike back and remain in that state either until the engagement isn't possible anymore or the causalities reach 50% of the unit men. In the first case, a transition is made to idle state, in second case the request reinforcement effector is called and the transition is to the waiting reinforcements state.

4.1.5 Waiting reinforcements:

This state groups two sensors: causalities (Have I lost half of my men?) with affirmative answer and waiting reinforcements (Have the reinforcements arrived?) with negative answer. The agent will remain in that states until its contingent have been refilled. Then it will change only the first answer and return to idle state. This happens because since the reinforcements have arrived, we are not expecting reinforcements anymore, so the answer for the second sensor will keep being negative.

4.2 Enemy Delay Finite State Machine

Enemy delay is used when the actual defense force is small and a large defense is been prepared on the background to engage the enemy soon. The goal is to delay the enemy providing time to the large defense force arrival. In that behavior units can occasionally do a retreat to the next planed position of defense.

All states in that behavior are the same as the previous one except for the waiting reinforcements state (Have the reinforcements arrived?) that will be replaced by retreating state (Have I reached my new position?).

4.2.1 Retreating state:

It groups two sensors: causalities (Have the reinforcements arrived?) and new position reached (Have I reached my new position?). The first sensor has affirmative answer while the second has negative answer. When casualties

Agent Type	Sensors	Effectors
Position defense	Can I see some enemy ?	Look for threat,
	It's a new target ?	Request Fire,
	Am I under artillery fire ?	Calculate enemy position,
	Am I under direct fire ?	Engage enemy,
	Can I engage the enemy ?	Request reinforcements
	Have I lost half of my men ?	
	Have I reached my new position ?	

Table 2: Table showing sensors and effectors for both kind of unit models.

reach 50% of the contingent it will retreat. Once the unit reaches the new position it will change only the first answer to negative and goes back to idle state. Again, once the position is reached, the unit will not be retreating anymore, so the answer for the second sensor will keep being negative.

4.3 Other automatic defense system features

4.3.1 Logistic

When defending an area where the goal is to delay the enemy, the terrain is divided in three lines parallel to the beach. The defense force hides provisions along the first two lines and takes position at the last line (closer to the beach). When the force can't sustain its line of defense it retreats to the previous line refilling its units with the hidden provisions.

In this paper we assume that the terrain is well known by the defense force and that the provisions such as ammunition, weapons, food and water are hidden in the lines of defense. When is necessary to retreat, the units are refilled with their initial provisions values when they reach the next position.

4.3.2 Periodic Aerial Reconnaissance

If the command post has some aircraft with no mission requested it will occasionally request an aerial reconnaissance mission that consists in a patronized fly over an area to plot any enemy activity. If any enemy is spotted it will include the position on the target list.

5 CONCLUSIONS

Applying artificial intelligence in military simulation systems may improve "players" interest over the training and reduce among of "players" necessary to the simulation allowing force to play only its paper. But we should be careful when using artificial intelligence. An extremely complex AI may lead to simulation overflow and players may even notice it, and a weak AI may invalidate training.

Agent model is adequate as an AI solution for military simulation systems. The great challenge is how to model efficiently an agent behavior. Agents provide an efficient way to model autonomous military units. Undoubtedly, for conventional forces, the best and easiest way to model behavior is the predefined behavior. All the necessary and exhaustive thinking that prevents programmers to apply this kind of behavior is already done and documented in the military strategic manuals. Small predefined rules can lead to complex behaviors and are easier to understand.

When working with no conventional forces predefined behavior may work, but it isn't perhaps the best choice because unconventional forces can act in an unexpected way.

It's very difficult to position units strategically without knowing the particular characteristics of the terrain. The first step to develop a tool to position units over any given terrain is to develop a way to identify the terrain characteristics.

6 FUTURE WORKS

Until now our units are manually positioned on the field by the defense instructors. We manage to develop a new module that allows the units to position themselves strategically over any given field.

We are studding the possibility of insert non deterministic behavior to model defense units that act differently from conventional combat forces, such as guerrilla forces.

Acknowledgments. The authors would like to thanks to Brazilian Marines Corps for the knowledge shared during this work and for the profitable years of partnership.

References

[1] J. David. *AI for Games and Animation: A Cognitive Modeling Approach*. A. K. Peters, 1999.

[2] N. K. Jaiswal. *Military Operations Research Quantitative Decision Making*. Kluwer Academic Publishers, 1994.

[3] C. J. A. Jr. and A. V. Gafarian. *Modern Combat Models: A Critique of their Foundations*. Operations Research Society of America, 1992.

[4] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995.

[5] R. B. Seixas and G. H. S. O. Lyrio. Riverine operations: Modeling and simulation. In *Advanced Simulation Technologies Conference – ASTC, in Military, Government, and Aerospace Simulation Symposium*, pages 33–39, 2004.