Simulation for Coral Reef Environment

NELSON MARCOS, REX RAYMOND FLORES, MICHEL ONASIS OGBINAR,
JOHN RICHARD SERA JOSE, JUAN JASON TABORDA
Software Technology Department
De La Salle University
2401 Taft Avenue, Manila
PHILIPPINES

Abstract: - A prototype for simulation of coral reef environment is presented. The simulation allows users to create a virtual coral reef environment which include marine lifeforms, at different trophic levels, represented as agents. These agents dwell and evolve within the coral reef ecosystem. The prototype simulates the interaction among agents which include human beings, and carnivores, herbivores, and producers in coral reefs. Effects of interactions among agents are recorded and reported. With a simulation software that shows such effects, reef managers will be able to predict such eventual outcomes and make sound decisions for the ecosystem. The multi-agent system uses rule-based algorithms in the interaction model to model consumption, growth, movement, and interaction.

Key-Words: - coral reef, ecosystem, multi-agents, rule-based algorithm

1 Introduction

Coral reefs, also referred to as rainforests of the sea, are very important animal habitats and their preservation are necessary to allow sea creatures to thrive in abundance. Various marine lifeforms such as fishes, corals, sea urchins, sponges, sea horses, sea anemones, star fishes, among others, depend on coral reefs to exist. However, a variety of assaults are taking their toll on these complex and delicate ecosystems. Most of these assaults, unfortunately, are caused by no other than human beings themselves. [1] [2]

To help promote the preservation of coral reefs, effects of various human activities on coral reefs should be made known to the public. One way to show people about the effects of these activities is through simulation software. This paper presents a simulation software designed and implemented to show the effects of man-made activities on coral reefs aside from the natural evolution and interaction among marine lifeforms. [3]

2 Coral Reef Problems

Several human activities harm coral reefs. *Overfishing* is one of them. A lot of people in the world rely on fishing as a means of livelihood and a lot people consume marine-based food. However, fishing in extremely large quantities should be discontinued. [4] [5] [6]

Illegal fishing, another problem, includes dynamite fishing, harvesting prohibited species, fishing out of season, fishing without permission, using outlawed types of fishing gear, and others. While many artisanal fishing techniques are limited in scope, a glaring exception is muro ami, a technique in which weighted bags are used to smash coral reefs to scare fish from their hiding spots.

Tourists, divers, and inhabitants near the coral reefs can also harm the reefs. Just mere touching corals or other marine lifeforms can destroy them. Fish collecting for the aquarium trade often involves poisoning reefs with sodium cyanide and other disastrous ingredients. Coral collecting is threatening some species with extinction. Moreso does littering and other manners of polluting the reefs. [7] [8]

Vessels, and structures built over and near the reefs also destroy the habitat of various marine lifeforms. Boat anchors can bump against the coral and break it. Beyond these attacks are the more ubiquitous threats from land-based pollution, sewage and sediment run-offs, dumping of waste, dredging, as well as from oil spills and other marine pollutants that degrade the water. [2] [9]

3 Simulation Design Framework

The simulation application, named DAGAT, is a *multi-agent* based system. Agents used in the system are representations of *artificial life*. [3] The agents are virtual representations of actual aquatic organisms, with features and actions that resemble these species.

The system was designed to have two main components, namely the *reef manager* and the *utility manager*. The reef manager handles all simulation agents, the spatial area where they reside, and other items found in the system such as environment cell types, as shown in Figure 1. The utility manager handles the analysis part of the simulation as shown in Figure 2. The *reef analyzer* acquires and translates the data from the reef manager and converts them into graphs and charts usable for interpretation by users. The *file manager* converts data gathered from the reef manager into a binary file, for storage purposes. The *file output writer* logs the events that occurred in the simulation and saves them into a text file. The *map editor* allows the user to customize the environment of the simulation, in order to create scenarios for better analysis. [3]

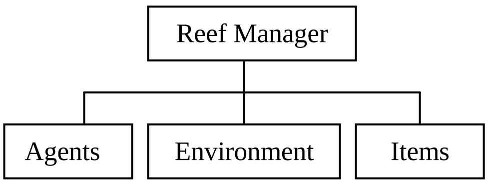

Fig. 1 Reef Manager Component

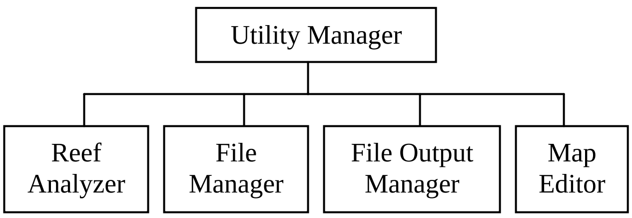

Fig. 2 Utility Manager Component

3.1 Agent Classes

All agents except for the trash fall under one archetype, the *agent class*. The class contains majority of the agent functions, such as *movement*, *growth*, and *consumption*. The agent class is subdivided into *fish*, *humans*, and *producers* as shown in Figure 3. [3] Fish include *herbivores* and *carnivores* while producers include *seaweeds* and *corals*. Producers in the system have no movement functions, only growth. Human agents in the system have a unique attribute called the *catch*, which determines the amount of fish agents it can capture in one run. Capture here is defined as the consumption of fish agents by human agents, which in turn reduces their number. Fish agents are similar to human agents, without the catch attribute. Fish agents move, grow, and consume their respective food supply, which is determined by their subtype.

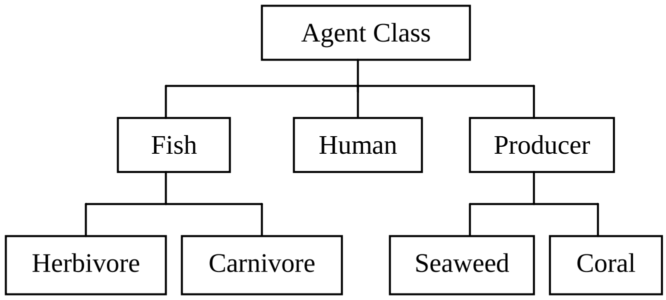

Fig. 3 Agent Class and Subclasses

3.2 Environment and Item Classes

The environment of the system is composed of *matrices*, which are composed of multiple *cells* as shown in Figure 4. Each cell has an identification and a limit or capacity pertaining to the number of agents that can stay in the cell. The environment contains functions for retrieving agent types in the system, as well as for generating the matrices for the respective agents. [3]

Figure 4 also shows the types of cells found in the matrices. Cells can be any of five types: *space*, *sanctuary*, *land*, *barren*, and *deep sea*. These five types differ in what kind of agent types are allowed in the cell. Space cell types can be occupied by all agent types. Sanctuaries allow any agent to access it, except human agents which are not allowed. Land cell types do not allow any agent to occupy the cell. Barren cell types can be occupied by all but producer type agents, which include corals and seaweeds. Deep Sea cell types also cannot accommodate producers, but has an effect on the consumption rate of agents.

Items Class in Figure 1 consists of other inanimate objects like trashes thrown into the reef.

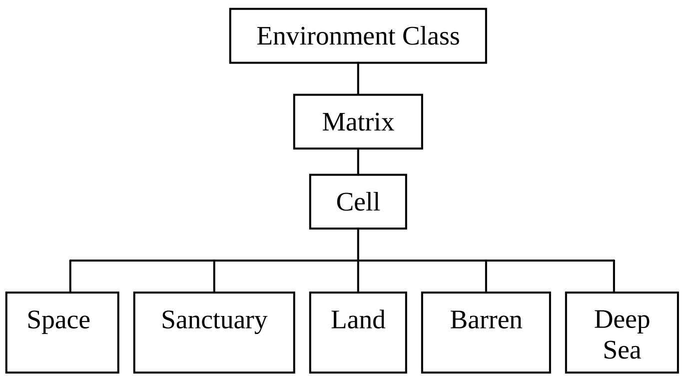

Fig. 4 Environment Class

The environment is represented using a *four-layer matrix*, each matrix inhabited by a particular agent type. The floor space matrix contains the producer agents which include the corals and seaweeds. Herbivore agents and carnivore agents are stored in separate matrices. The topmost layer contains the human agents. Each matrix is composed of a 200 x 200 array of cells. Each cell has a biomass capacity for setting the amount of agents in a given cell, as well as a cell type. The cell type determines what types of agents are allowed in the cell. Some cell types forbid any agent access, while others only limit certain agents to pass. Table 1 shows the summary of cell types and agents allowed in the cell types.

Table 1: Summary of Cell Types

Cell Type	Allowed Agents
Space	ALL
Barren	Herbivore, Carnivore, Human
Sanctuary	Producer, Herbivore, Carnivore
Land	NONE
Deep Sea	Herbivore, Carnivore, Human

3.3 Agent Functions

Agents in the system *interact* with one another to fulfill their goals, and to achieve certain objectives. There are many agents in the system, each type performing certain actions that are different from others. The functions in the subsections are some of the major aspects or actions the agents perform while running the simulation. [3]

3.3.1 Movement

Agents capable of movement in the system use two functions for movement in the matrix. First, the function *suggestMove()* is called, which implements the *movement rules* of the system. Movement decided in the system differs between human agents and fish agents. Human agents decide on their next destination solely on the *proximity* of the next cell with available catch. The human agent checks all cells which he can reach in a given time. The available *catch* and *distance* are also taken into consideration, after which the human agent moves to the nearest available cell with fish to catch.

Fish agents employ a different *rule-based algorithm* for deciding to which cell to move. The fish agent uses the same suggestMove() function, but the weights used to determine the most appropriate area to move to are different.

The weights compared are the *threat risk* and the *amount of food* available in the cell. Threats are given more weight, and the value for threat is determined by the difference between the biomass of the fish agent and the biomass of the agent causing the threat. The other weight computed is the amount of food available for the agent. This is computed as the comparison between the biomass of the agent, and the biomass of the prey or food. When both of these factors are taken together, the total weight of the cell is produced. The cell with the highest number will be chosen as its next location.

3.3.2 Consumption

Some agents are designed to consume agents of different types. Each agent consumes another type of agent, with the exception of the producer, which does not have a *consume() function*. Consumption of an agent reduces its number by a given amount. If the agent number is reduced to zero, the agent is deleted by the system, after calling the *kill() function*.

A function called *checkForPrey()* is used by agents to check the adjacent agents if these can be consumed by the current agent. No single agent can consume all agents. There are limits to what type they can consume, and this function makes sure that the agent consumes only those that it is allowed to consume.

3.3.3 Growth

Agents continuously grow in number, specified by a *growth rate* preset. When a cell reaches the maximum capacity of agents it can sustain, the agent cannot grow anymore, because additional number of agents will not fit into the cell. Instead, a new agent will be created at an adjacent cell, and the supposed additional agents will be placed in the new cell. This is the concept of *spilling*.

Birth is abstractly represented by the *growth* of the agents. The agent's growth of biomass also already takes into account the amount of biomass an agent may give birth to (new agent birth is infused in an existing agent's growth). If an agent exceeds a cell's biomass capacity, a new agent will *spawn* at an adjacent cell with a biomass equal to the excess agent's biomass. Thus, *spawning* may occur during spilling.

3.4 Human Agents

Human agents represent the human population living off the reef fisheries. This class is an extension of the agent class, but with different roles. Human agents have a *setCatchBiomass()* function, which determines the amount of fish it can catch at a given time. The human agents also have a cycle, which determines its catch size, the movement it will take, and a function to catch fish agents. Human agents are further characterized by the *fishing method* they employ. [3]

The fishing methods affect how the fish are caught, and how they affect the nearby environment. There are four fishing methods, namely: *normal fishing*, *dynamite fishing*, *cyanide fishing*, and *muro-ami*.

Normal fishing attempts only to remove the fish in the same cell as the fisherman. Dynamite fishing not only attempts to catch fish, but the fisherman also *bombs* the cell, destroying a portion of all the agents with the same coordinates specified by the *blast radius*. These agents include seaweeds, corals, and herbivores, which are not included in what the human agent consumes.

Cyanide and muro-ami fishing methods also have similar effects, but have different degrees. Cyanide fishing kills seaweeds and corals within the *poison radius* and kills fish more than the amount that can be fished. Muro-ami kills seaweeds and corals the net can reach and kills fish more than the amount that can be fished.

4 Simulation Prototype

This section shows some of the features included in the design of the simulation prototype for coral reef environment. [3]

A map editor is provided to allow the user to create a map and specify the cell types, agents, and agent rates (e.g., growth rate).

The system was designed to allow users to start a new simulation, load existing simulation, and save simulation.

The current simulation can be controlled easily with play, pause, unpause, stop, and fast forward. Simulation is by month. Using the fast forward control, the user can specify the number of months to jump.

Percentages of the number of agents per type (seaweed, coral, herbivore, carnivore, and human) is shown as toolbars which updates during the simulation.

Statistical as well as graphical results of a simulation are provided for reef managers to use. *Population sizes* (in *biomass*) as well as *average fishermen catches* were recorded as results of the simulation runs.

Figures 5 to 10 shows various screen shots from the simulation prototype. Figure 5 shows the agent and cell types used in the system. Figures 6 and 7 shows simulation options and controls available for the reef manager.

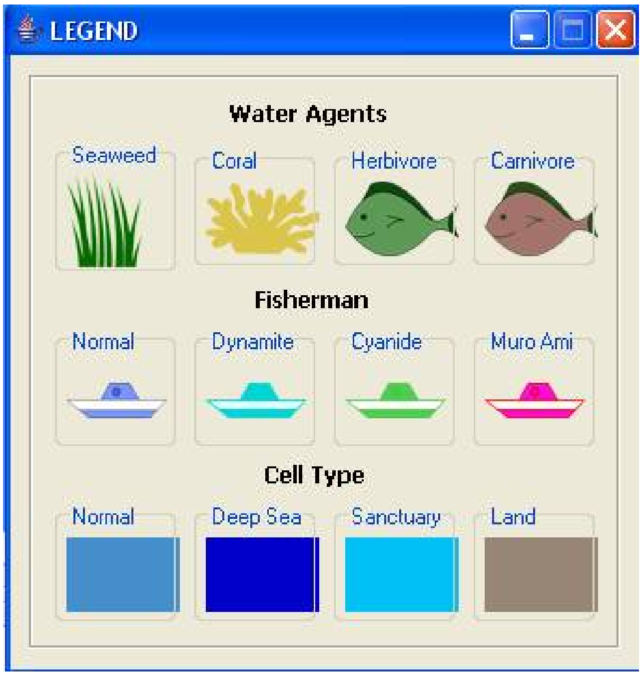

Fig. 5 Agent and Cell Types

Fig. 6 Simulation Options

Fig. 7 Simulation Controls

Figure 8 shows a screen shot of the map editor, Figure 9 shows a sample screen shot of a simulation, and Figure 10 shows a sample graph of a simulation result.

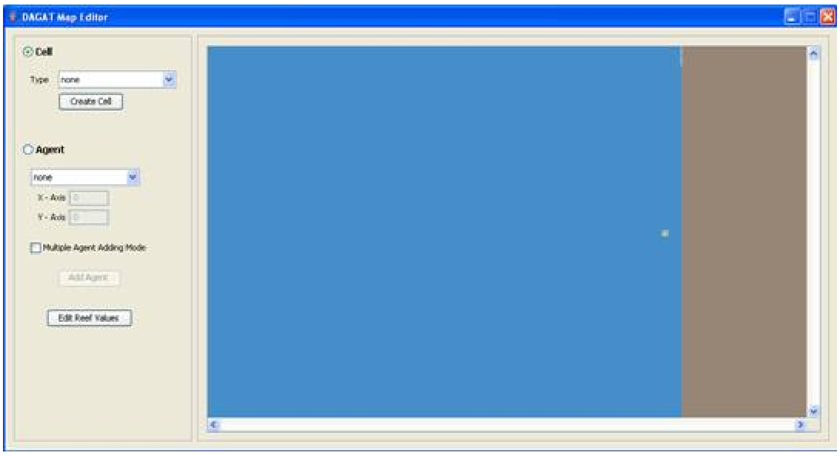

Fig. 8 Map Editor

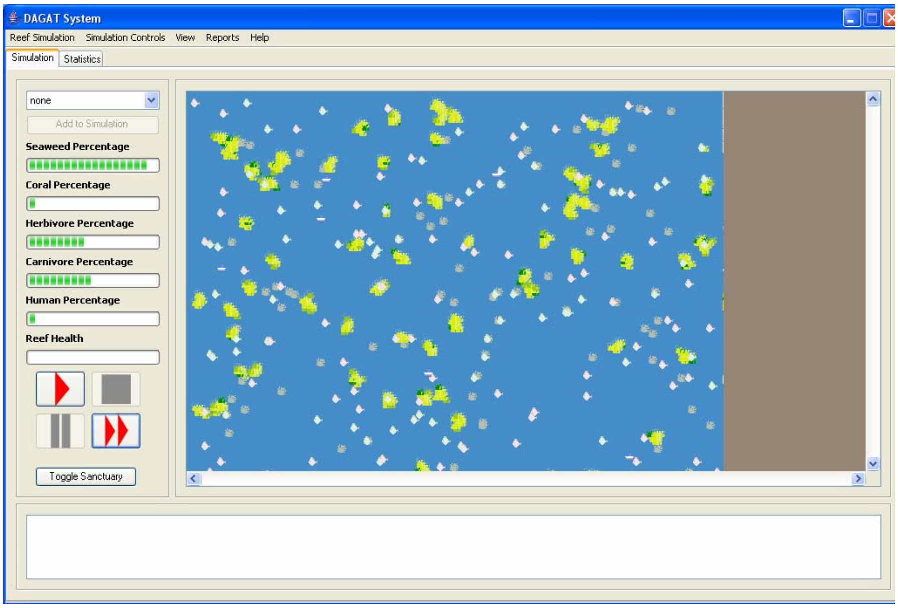

Fig. 9 Sample Simulation

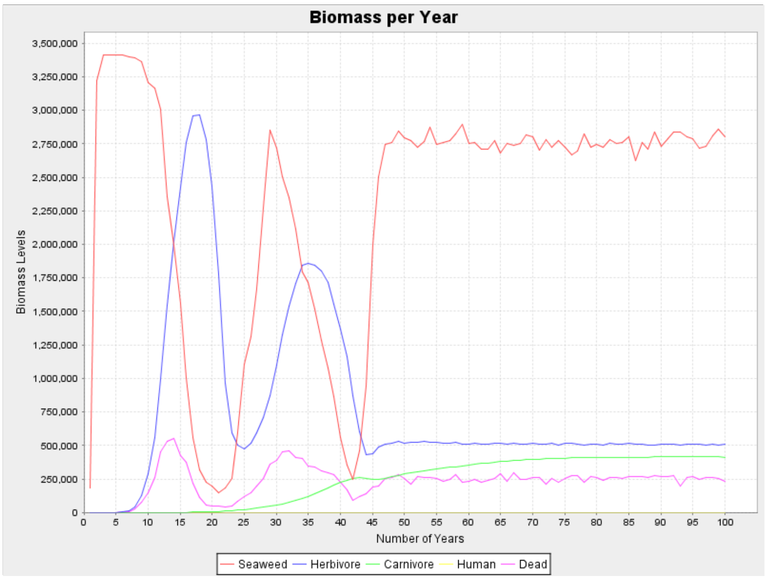

Fig. 10 Sample Graph of a Simulation Result

The simulation prototype was tested with various cases to verify the validity of the implementation. Various combinations of the different types of agents (herbivores, carnivores, seaweeds, corals, and human beings) were used in different simulations to be able to verify correctness of the formulas used, constants and variables used, modeling of the interaction of agents, among others. The simulation results were verified with experts and consultants on coral reefs.

Data results from the tests show that the simulations are more stable when the amount of producers is greater than the amount of fish agents, regardless of the amount of humans. [3] When the amount of fish agents is equal to or lesser than the producer agents, the simulations are more likely to fail. The number of human agents in the system gave a significantly less effect on the simulation, compared to the ratio of the fish agents and producer agents.

The ideal graph results for agent population would be an oscillating flow, meaning that no population size is consistent. [3] Each agent type would increase and decrease periodically, in an effort for the system to stabilize. A simulation produces accurate results when such an effect occurs.

5 Conclusion

The prototype developed is a population-wide simulation system in which four trophic levels of biomass balance are included. Modeling the various agents, their properties, and interaction results in a simulation system which is rather complex. Speed was encountered as an issue while running the system because of the numerous agents interacting, as well as the graphics involved.

The simulation prototype also modeled growth rates, consumption rates, and other delicate factors to balance the reef. Temporal factors considered also added to the complexity of the system.

Even though the system developed seems complex enough, it is not enough to completely model a coral reef environment. Ecosystems are naturally very complex which leads the proponents to conclude that the results in the simulation experiments are approximations to what might actually happen in a coral reef environment given certain scenarios.

Future work can concentrate on other agents and agent behaviors that can be modeled and other factors that affects a coral reef environment. Further improvements on spatial and temporal aspects of the simulation can also be done for speed considerations.

Acknowledgement
The researchers would like to acknowledge the great help of the consultants for this project – Dr. Wilfredo Licuanan (DLSU Biology Dept) and Mr. Rollan Geronimo (UP Marine Science Institute).

References:

[1] Pauly, D., Ecosystem impacts of the world's marine fisheries, *Global Change Newsletter*, 55, 2003, p. 21

[2] Solandt, et al., Reef Fish Populations around Danjugan Island, Negros Occidental, Philippines, 2002

[3] Flores, R., Ogbinar, M., Sera Jose, J., Taborda, J., & Marcos, N., DAGAT: DAnjuGan Artificial environmenT, Implementation of a Coral Reef Simulation, De La Salle University, 2006

[4] Sale, P. Ecology of Fishes on Coral Reefs, 2nd Ed, Academic Press, San Diego, CA, 2002.

[5] Choat, J. H., & Bellwood D. R., Reef fishes: their history and evolution, *The Ecology of Fishes on Coral Reefs*, Academic Press, San Diego, CA, 1991, pp. 39-66

[6] White, A., Alino, P., Meneses, A., Creating and Managing Marine Protected Areas in the Philippines, *Fisheries Improved for Sustainable Harvest Project*, Coastal Conservatation and Education Foundation, Inc. and University of the Philippines Marine Science Institute, Cebu City, Philippines, 2006, p. 20

[7] Klenk, Marine Ecology, [online], http://darter.ocps.net/classroom/klenk/Ecology.htm, 2005

[8] Marine Life: Marine Ecosystems [online], http://www.biosbcc.net/ocean/marinesci/03ecology/mlecosystem.htm

[9] Geronimo, R., Uychiaoco, A., Lim, M., Seymour, R., Bantang, J., & Gomez, E., Assessing the impact of biodiversity changes on reef fisheries using an individual based model, 2005