AGENT-BASED MODELING SOFTWARE FOR NATURAL AND RURAL ECOSYSTEMS

Natalia Bogach, Vadim Dyachkov, Anton Lamtev, Yurij Lezhenin
Peter the Great St. Petersburg Polytechnic University, Russia
bogach@kspt.icc.spbstu.ru, dyachkov@kspt.icc.spbstu.ru, lamtev@kspt.icc.spbstu.ru
lezhenin@kspt.icc.spbstu.ru

Abstract. Agent-based modeling in biology allows, firstly, to simulate a multi-component environment with a number of entities (plants, microbes, fungi, insects, birds, mammals etc.) and, secondly, to perform numerical experiments for different research purposes. The article presents modeling software for natural and rural ecosystems based on the food chain concept. This software encompasses C++ core and Jupyter/Python data analysis environment; it supports both an interactive and a batch run modes; it allows exporting data for further processing and visualizes the data analysis results. Agent features and motion strategies can be defined with the program interface. The software allows modeling ecosystem dynamics with respect to inter- and cross-species relations through the simulation of agent analytical models in ecosystem world. We applied a three-species test model "plant-herbivore-carnivore" (PHC model) to perform software code validation. As a testing scenario, foraging impact to the overall PHC model balance was evaluated during the series of software try-outs.

Keywords: agent-based modeling, food chain, foraging, ecosystem balance.

Introduction

Industry and agriculture are the promising areas for artificial intelligence (AI) software. Among the other AI technologies, agent-based modeling is widely used to analyze complex natural systems. A food chain concept is frequently applied for modeling the relations in the ecosystem where components depend on each other. Agent-based modeling provides a powerful tool to explore ecosystems by means of numerical experiments. It involves simulation of complex ecological phenomena and is based on analytical models of ecological processes [1]. An ecosystem model helps to gain understanding of a real system [2], i. e. to reveal the interactions inside the processes in order to be able to plan or predict ecosystem behavior as well as to evaluate the impact and result of specific activities [1; 3; 4]. Ecosystem modeling can be applied in a wide range of biological science areas, e. g., natural resource management, environmental health and sustainability, agriculture, wildlife protection, etc.

An appropriate choice of a modeling tool is a crucial issue. Searching for the modeling software we browsed through a variety of tools and frameworks. Currently, there exist a number of the special multi-agent modeling frameworks for ecosystems at four levels of abstraction.

1. Extensions of single-agent models taking into account only a few specific interactions [5].
2. Minimally Realistic Models which represent a subset of a certain ecosystem [6].
3. Holistic ecosystem models that attempt to simulate all levels in the ecosystem. *Ecopath with Ecosim* (*EwE*) [8; 9] or *GoldSym* [10] are the examples of the widely used multi-level ecosystem models.
4. Dynamic system models representing both bottom-up (physical) and top-down (biological) processes acting in an ecosystem, e. g. *Atlantis* [11] – a dynamic system for marine research.

At the same time, general purpose modeling frameworks are able to perform ecological simulation as well. The following requirements have been elaborated for the objective and unbiased framework estimation and comparison:

1. body/behavior agent description;
2. general purpose programming language;
3. open source license;
4. current maintenance;
5. batch run mode;
6. data analysis support;
7. low barrier to entry.

Among the others AnyLogic [12], GAMA [13], LSD [14], MadKit [15], MESA [16], RePast [17], were of the special interest because they satisfy almost all the requirements listed above. Despite the possibilities provided by all of the ready-made frameworks they still require coding of the model

DOI: 10.22616/ERDev2017.16.N150

components and agent life processes. At the same time, a developer does not get any independence from the framework project infrastructure and implementation specificity.

Another way is to develop a custom model simulator from scratch, which is challenging, but gives all the freedom and flexibility to a researcher. The latter was convincing enough to encourage the authors to implement a custom ecosystem model simulator. Even though we decided against the usage of popular modeling tools for different reasons, our software tool is aligned with their best practices. It has two main parts, namely, C++ model core and Jupyter/Python data analysis environment. This architecture makes possible to perform a batch run and export data through command-line interface for a large number of numerical experiments in biology using an ordinary computational device.

The rest of the paper is structured as follows: Section 2 describes the architecture and components of modeling software, it also introduces the PHC model used for the software code validation; Section 3 provides the software try-out results and discusses the applicability of modeling software proposed, Section 4 concludes the paper.

Materials and methods

Ecosystem modeling software architecture and components

Program core (Fig. 1) allows the researcher to implement the ecosystem features and agents at his choice, e. g., animal's decision making logic and moving strategy.

Data acquisition is performed through Command Line Interface (CLI). C++ implementation was motivated by the high performance and efficiency of the language. The source files of the model can be accessed on GitHub and can be compiled. The data analysis part is deployed on Jupyter, an interactive data science and scientific computing web-application. Launch parameters, data analysis and visualization of the results for tests are set by Python scripts in Jupyter notebooks. All the components of the experimental infrastructure and original artifacts are available and the results can be easily repeated, replicated or reproduced [18]. The Docker container on DockerHub is provided [19].

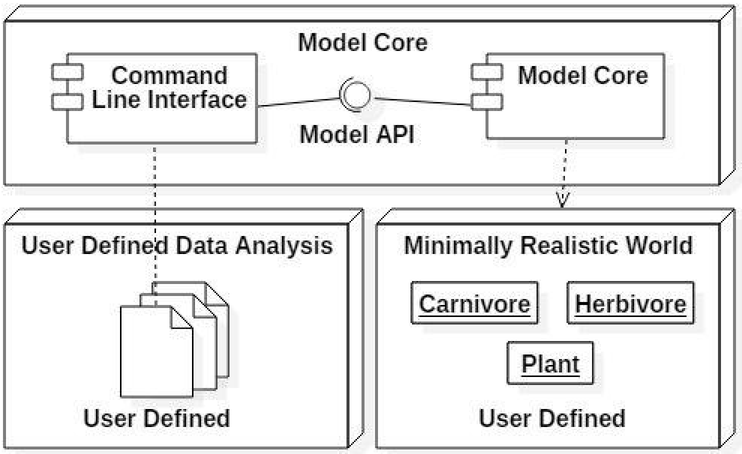

Fig. 1. M**odeling software architecture**

"Plant-herbivore-carnivore" model (PHC model)

Software functional tests were performed in the test environment. For code validation purposes we defined a three-species ecosystem connected by a food chain "plant-herbivore-carnivore". The software code was validated through simulating this test model.

PHC model is constructed to have two levels of abstraction: at the single-agent level it operates agent models, which include agent parameters, namely, age, speed, vision, reproduction rate, energy and stamina. Animals are driven by their "brain", i. e. they have four types of behavior and make their decisions based on the energy level and the environment state. At the second level we construct a minimally realistic world, defined by a set of macro-parameters, namely, field size, number of agents of each type and plant spawn interval.

Single-agent model

Animals are described by a set of temporary evolving parameters: age, speed, life energy, stamina and vision. At the same time, the plants are defined only by age. A number of steps t is taken as an independent argument, while the model parameters depend upon it directly or implicitly.

The age A of an agent depends only upon a number of steps. Animal's life passes through 3 phases: youth, middle age and old age. The speed is a distance covered during one step. Animal speed is limited by a maximal speed threshold V_{max}, which depends upon the age and the baseline speed level V_0. It can vary from 0 to maximum depending on the current situation in the field. The carnivores and herbivores ability to move is limited by their stamina S. If the stamina is equal to 0 the agent cannot move and needs to have a rest. The stamina changes at each step. Vision is a circle around the agent with the radius R and it depends upon the age as well.

PHC model behavior

Agents have "brain" and make decisions, while the plant behavior is straightforward: a fixed plant amount appears at random positions within a fixed number of steps. At each step an animal makes decisions based on the target list. Every target on the list is a structure {a target type, distance to a target}. In this way each animal gets a picture of the environment around him. The decision making system comprises walking, eating, reproduction and pursuit/runaway behavioral patterns (Fig. 2):

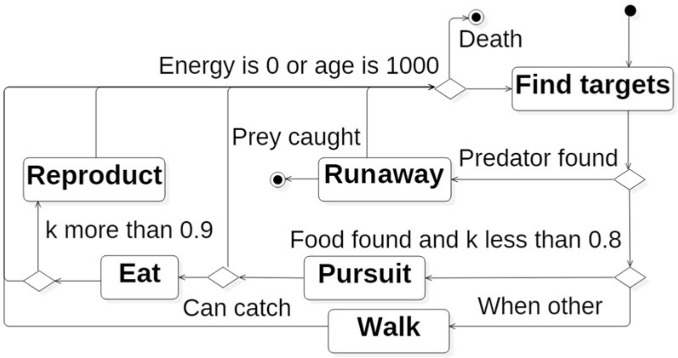

Fig. 2 **Animal brain**

- Walking is a free agent moving. During walking the animal expends its moving energy and restores its stamina.
- Eating is a food consumption process that gives the energy. We assume that eating is performed within a current step, i.e. instantly.
- Reproduction is adding a new animal into the model. After that the energy level of the parent is reduced.
- Pursuit or runaway is moving with the maximum available speed towards the herbivore or from the carnivore along the line connecting their initial positions. During the pursuit or runaway the animal expends the moving energy and loses its stamina.

Test model world

The world W of the test model is defined by macro-parameters $\{\{F_{2D}\}, N_h, N_c, N_p, T_p\}$, where $\{F_{2D}\}$ is a field size, N_h, N_c, N_p are numbers of each type of the agents measured in field cells occupied and T_p is plant spawn interval measured in simulation steps.

World simulation is performed in a rectangular closed area divided into cells. Each agent occupies one cell. It moves to a number of cells depending on its speed. The test model evolution history is observed during simulation. The lifetime of the test model world W is measured as a number of steps during which the model stays alive.

Results and discussion

Software try-out scenario was designed to find out whether the software can properly show the influence of one parameter of the PHC model to the others. Therefore, as a test, we explored foraging dependences in this model and observed how its lifetime and animal population waves depended upon a foraging schedule. The foraging schedule was denoted by the amount of plants N_p appearing at the specific moments separated by the plants spawn interval T_p.

This simulation was performed under the following conditions:

- The field size {F_{2D}} was 100x100 cells.
- The animal lifetime limit was set to 1000 steps.
- The first generation of the animals started to produce its offspring after 400 steps.
- The initial energy level was set to a starting value E_0 and held on for 50 steps to keep the young animals alive.

The software try-out was performed in two steps: the goal of the first step was to search for a subset of test model macro-parameters {N_h, N_c, N_p, T_p}, which gave the longest model lifetime; the goal of the second step was to evaluate the foraging schedule impact to the population waves of the test model under the macro-parameters found at the first step.

Lifetime maxima search

The goal of this test was to search for macro-parameters that gave the longest model lifetime. We launched 3000 cycles of modeling for various PHC model initial conditions {N_h^i, N_c^i, N_p^i T_p^i}. For all local maxima foraging intensity N_p/T_p varied (Fig. 3).

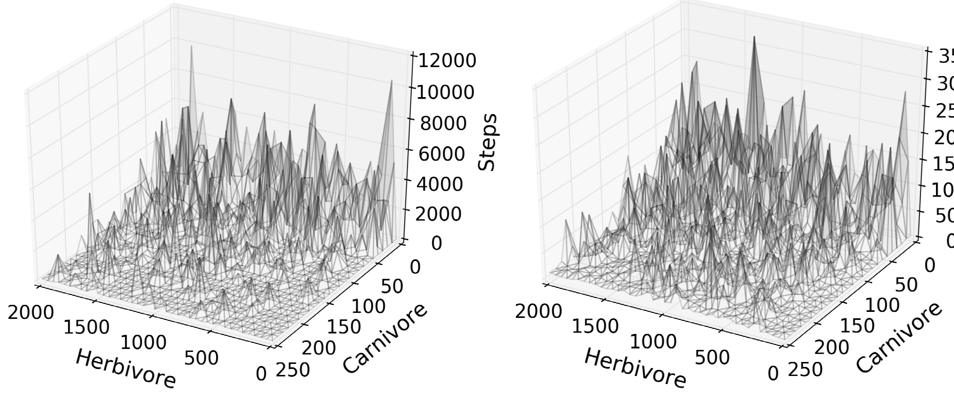

Fig. 3. **Lifetime maxima area expansion under increasing foraging**

The results show the similar behavior in a wide range of initial settings. One can notice that all the points {N_h, N_c, N_p, T_p} where the lifetime local maxima have been observed are located in the vicinity of each other (Fig. 3), but the increasing of N_p results in expansion of the area covered with peaks. The subset of macro-parameters $\{N_h, N_c, N_p, T_p\}_{i=1}^{P}$ that return the lifetime peaks has been stored and used at the next stage of try-out.

Foraging schedule impact to amplitude and period of population waves (population dynamics)

At the next stage we evaluated how the foraging schedule affects the population waves. It was obvious, that foraging determines the size of the herbivore population. The software was supposed to testify in favor of this fact. We chose the initial settings from a subset of macro-parameters found at the first stage $\{N_h, N_c, N_p, T_p\}_{i=1}^{P}$ and estimated how foraging schedule changed the population waves amplitude and period. In Fig. 4 (a) the amount of plants N_p is sufficient for the current value of N_h, so the population growth is stimulated. The simulation starts at around $N_h = 3000$ and overcomes this value at the following population waves maxima. On the other hand, in Fig. 4 (b) when the initial number of the plants is poor, the herbivores population rapidly decreases during the first several steps and never reaches the initial level.

We also observed that the amplitude of N_h, grew in accordance with the amount of plants given to the system N_p and the oscillation period of N_h correlated with the plants spawn interval T_p. The long spawn interval produced the larger oscillations; even when the amount of plants was sufficient, it did not compensate the lifetime decrease.

To evaluate the foraging schedule effect upon the test model lifetime we fixed the number of animals and launched our model with various foraging parameters. We repeated this test for various N_h

and N_c ratios and observed similar outcomes. One can see that the ridges of lifetime maxima are stretched along the line corresponding to an approximate foraging intensity of $N_p/T_p = 5/2$ (see Fig. 5).

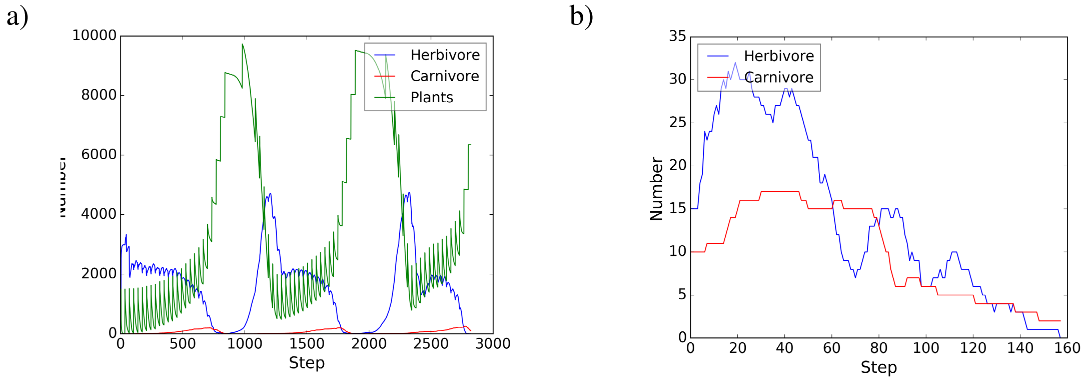

Fig. 4. **Population dynamics under sufficient (a) and poor (b) foraging**

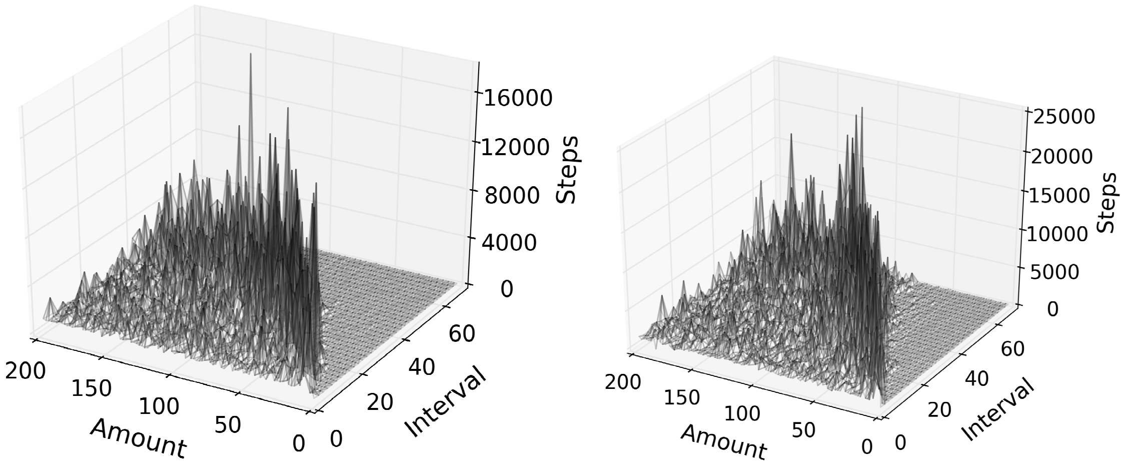

Fig. 5. **Lifetime maxima group along the line $N_p/T_p = 5/2$**

Plants act as a bounding and regulating factor in this ecosystem. The amount of plants limits the number of animals existing in the field. Therefore, whatever large N_h could be initially, it falls rapidly to a certain level within the first modeling steps. The increase of foraging given to the system causes the increase in the number of herbivores and, respectively, the number of carnivores. Thus, the area covered by the lifetime peaks expands under increasing foraging. Foraging intensity affects N_h and N_c waves amplitude and period, while under large T_p the overall model lifetime decreases. The lifetime decrease reveals the ecosystem equilibrium loss. On the other hand, smooth and uniform foraging has a better perspective for a longer ecosystem lifetime.

Conclusions

The software try-out results provide evidence in favor of the fact that foraging schedule is significant as far as the ecosystem lifetime is concerned. Thus, our software allows modeling and observing specific ecosystem phenomena in accordance with the expectations of the researchers. The food chain length, the number of species, agent features and motion strategies can be defined with the program interface. Our software allows modeling the ecosystem dynamics with respect to inter- and cross-species relations through the simulation of agent analytical models in ecosystem world.

References

1. Acosta H., Wu D., Forrest B.M. Fuzzy experts on recreational vessels, a risk modelling approach for marine invasions. Ecological Modelling, 221(5), 2010, pp. 850-863.
2. Hall C.A.S., Day J.W. Ecosystem Modeling in Theory and Practice: An Introduction with Case Histories. University Press of Colorado, 1977.

3. Bennett N.D., Croke B.F.W., Guariso G., Guillaume J.H.A., Hamilton S.H., Jakeman A.J., Marsili-Libelli S., Newham L.T.H., Norton J.P., Perrin C., Pierce S.A., Robson B., Seppelt R., Voinov A.A., Fath B.D., Andreassian V. Characterising performance of environmental models. Environmental Modelling and Software, 40, 2013, pp. 1-20.
4. Kelly R., Jakeman A., Barreteau O., Borsuk M., ElSawah S., Hamilton S., Henriksen H., Kuikka S., Maier H., Rizzoli A., van Delden H. I., Voinov A. Selecting among five common modelling approaches for integrated environmental assessment and management. Environmental Modelling and Software, 47, 2013, pp. 159-181.
5. Hollowed A.B., Bax N., Beamish R., Collie J., Fogarty M., Livingston P., Pope J., Rice J.C. Are multispecies models an improvement on single-species models for measuring fishing impacts on marine ecosystems? ICES Journal of Marine Science, 57(3), 2000, pp. 707-719.
6. Begley J. Gadget user guide. Marine Research Institute, Reykjavik, Iceland, 2005.
7. Punt A.E., Butterworth D.S., de Moor C.L., De Oliveira J.A.A., Haddon M. Management strategy evaluation: Best practices. Fish and Fisheries, 17(2), 2016, pp. 303-334.
8. Christensen V., Pauly D. ECOPATH II - a software for balancing steady-state ecosystem models and calculating network characteristics. Ecological Modelling, 61(3-4), 1992, pp. 169-185.
9. Pauly D., Christensen V., Walters C. Ecopath, Ecosim, and Ecospace as tools fore valuating ecosystem impact of fisheries. ICES Journal of Marine Science, 57(3), 2000, pp. 697-706.
10. GoldSym [online] [22.02.2017]. Available at: http://www.goldsim.com/Library/Models
11. Fulton E.A., Smith A.D.M. Lessons learnt from a comparison of three ecosystem models for Port Phillip Bay, Australia. African Journal of Marine Science, 26(1), 2004, pp. 219-243.
12. AnyLogic [online] [12.11.2016]. Available at: http://www.anylogic.com/agent-based-modeling
13. GAMA [online] [13.11.2016]. Available at: http://gama-platform.org
14. LSD [online] [15.12.2016]. Available at: http://www.labsimdev.org/Joomla_1-3
15. MadKit [online] [17.12.2016]. Available at: http://www.madkit.net/madkit
16. MESA [online] [15.01.2017]. Available at: http://mesa.sourceforge.net
17. RePast [online] [16.01.2017]. Available at: https://repast.github.io
18. Project on Github [online] [22.03.2017]. Available at: https://github.com/Zhuikov/predator-prey
19. Project Docker container [online] [22.03.2017]. Available at: https://hub.docker.com/r/zhuikov/predator-prey