PATTERN ORIENTED AGENT-BASED MULTI-MODELING OF EXPLOITED ECOSYSTEMS

Stephane Bonneaud, Pascal Redou, Pierre Chevaillier

Computer Science Laboratory for Complex Systems – LISYC
European Center for Virtual Reality – CERV
25, rue Claude Chappe, BP 38 F-29280 PLOUZANE, France

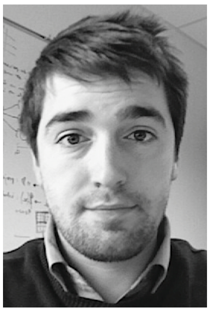
bonneaud@enib.fr (S. Bonneaud)

Abstract

Modeling and simulating exploited ecosystems is a complex process that most often requires to manipulate a complex model. Indeed, experts have to use heterogeneous models and assemble them to build the global model, possibly in an incremental manner. We argue that a multi-model methodology and simulation tool enabling to partly automate the modeling and simulation processes and help grasp the complexity of the real systems are needed. Based on the pattern oriented modeling and the multi-agent approach, our proposition is data-oriented. We use a society of models based on agents that interact through the environment and enable coupling of models through environment mediated influences. We specify three roles for modeling agents: (1) the *model-agent* handles an expert's model; (2) the *controler-agent* watches upon the environment; (3) the *observer-agent* builds observable objects. Goals of *model-agents* are structured by the inputs and outputs of the agents that have a specific semantic and shape. The environment is organized by the data (*artefacts*) in which *patterns*, produced by *model-agents*, are outlined. In this paper, we exhibit the framework and methodology of our proposition. We also try to show that our *model-agent*-based approach can help experts build their models in collaboration with agents and we exhibit the local processes that enable us to envision automation of the modeling process. At last, we use a didactic simulation scenario of a theoretical exploited fish population to exhibit our methodology and simulation tool.

Keywords: Modeling methodology, pattern oriented modeling, multi-agent simulation

Presenting Author's Biography



Stephane Bonneaud is doing its PhD in the Computer Science Laboratory for Complex Systems at the European Center for Virtual Reality, where he works on the modeling and simulation processes of exploited ecosystems. Stephane has a master degree (research) in cognitive sciences (Paris 11), where he had to deal with automatic extraction of semantic patterns in discussions. He also has a master degree (research) in Artificial Intelligence (Paris 6) where he worked on modeling and simulating emergency evacuation scenarios using agent-based techniques and working with the Ω language.

1 Introduction

We focus on the modeling and simulation processes of exploited fish populations in dynamic environments. The goal is to understand how environmental and economic phenomena impact the viability of exploited marine ecosystems. Beyond this object of study, the purpose is to build a modeling methodology and a simulation tool to handle complex models. In other words, experts have to manipulate different models, existing or to be stated, that need to be rationally assembled. Hence, simulation environments enabling the coupling of models of different natures are needed. But the challenge is also to partly automate modeling and simulation processes. Therefore, we propose a modeling and simulation framework enabling experts to handle model heterogeneity and autonomous and proactive behaviors for automatic building and analysis of models.

We suggest that the modeling process should be data oriented. When studying a system, experts have specific information at hand available with specific assumptions. They study the system through that information. We propose to base our modeling on the pattern-oriented framework (POM) [1, 2]. Model coupling is enabled, but models are meaningful because of the data they use and produce. On the whole, our proposition is based on the multiagent paradigm, using *model-agents* among other agents. Our *model-agents* handle models and interact via the environment supporting the *patterns* that structure the data of the experts. This solution formalizes the coupling based on influences between agents through the environment and is therefore a structural framework. But it also addresses the automation issue by structuring the modeling process around agents with identified roles and behaviors to control, perceive and act on the environment. In the following sections, we will first describe the modeling experience of exploited ecosystems, the POM and the issues we address. Then we will describe our modeling methodology exhibiting the entities and concepts of our system. Afterwards, we will exhibit our agent-based solution and, at last, we will use a theoretical simulation scenario to illustrate our methodology and simulation tool.

2 Context

2.1 Modeling exploited marine ecosystems

Marine biological populations and their exploitation by different fishing vessels form a heterogeneous, open and dynamic system. Composed by many dynamical and interacting entities, such a system has a hardly predictable overall behavior. Multi-specific in nature, fish populations are renewable shared resources in a changing environment influenced by climate change and economic markets. Fisheries, composed of different types of fleets, are weakly structured with no centralized governance. Interactions between all those components include predation and migration processes, spatial competitions, exploitation strategies and social phenomena.

Building and manipulating models of such a system is challenging. Modeling and simulation are needed when it comes to understanding the processes structuring those systems and eventually evaluating scenarios of the system's possible futures. Ecological phenomena are classically stated using differential equation systems. This approach enables the production of compact models explicitly expressing the system's behavior [3]. Another modeling option consists in directly considering the individual components of the system using an individual-based approach (IBM) [4]. The global behavior of the system is not explicitly stated anymore but emerges from the individuals' interactions [5]. Many IBM solutions exist addressing ecological [6, 7, 8, 9] or economic issues [10, 11]. But while it is difficult to express individual variability in the equation based approach, the IBM approach can request much processing power and produce models with inaccurate results [5, 12]. Depending on the question(s) the modelers want to address, heterogeneous modeling material might be needed in order to build the global model of the studied system. Moreover, exploited ecosystems are structured by phenomena which modeling might be performed by different experts coming from different fields. Those experts therefore need to build together a complex model composed a various sub-models, each of which addresses a specific phenomenon. We stress out that the global model has to be experienced by simulation but also during the modeling process itself. Experts have to be able to incrementally assemble those sub-models in order to partially experience and eventually validate them or even merely get a feeling of the functioning of their system. At last, producing simulation results requires to execute a structured, lasting and tough course of actions: testing the model under different conditions and scenarios, analyzing variables, carrying out sensibility analysis among others and organizing all the produced data and investigations. Therefore, we believe that the system should be partly automated in order to execute part of the work and help the experts in their modeling and simulating experiences. We try to address this issue by building a framework in which automation can be envisioned and we try to investigate the methods and tools that could be used for multi-modeling such complex models.

2.2 Pattern oriented modeling (POM)

A pattern is the observation of an emergent non random structure. It is, for GRIMM, the indicator of fundamental structural processes and components of a studied system. It contains and exhibits information on the system's internal organization and on the mechanisms that structure it [1]. *Patterns* are pieces of information that denote a specific behavior of the system. The pattern oriented modeling (POM) [1, 2] was proposed to make the modeling process more rigorous and comprehensive. Modeling must be driven by one or several issues on a clearly identified object of study. Questions and assumptions on this object are identified by remarkable features or *patterns*. The POM states that the structural model of the system should be built on those observed *patterns*. The model is therefore directly linked to the existing organization and structure of the

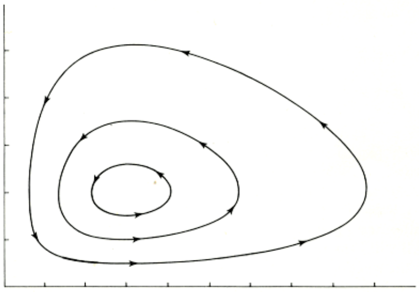


Fig. 1 Example of a *pattern* - The preys' population size is plotted versus the predators' population size.

real system [13]. Figure 1 shows a typical pattern of a prey-predator system. *Patterns* are often key variables of the system plotted together. In the case of the prey-predator system, we don't plot the number of preys in time or the number of predators in time, but we want to exhibit the interaction between the two populations by plotting the preys' number and predators' number together. More generally, *patterns* can also be spatial structures identified in spatialized data like maps, for instance, population migrations and distributions. We often manipulate such *patterns*.

3 Material and method

3.1 Modeling environment: *patterns* and *artefacts*

Based on the POM methodology, modeling becomes data oriented and the mechanisms or phenomena chose to structure the model are selected for their capacity to explain the *pattern* (s). We therefore built a modeling framework organized around the *patterns* explicitly described and considered as key components of our system. As seen in the example above, a *pattern* is often a composition of some other data. *Patterns* are not always directly produced by models but some post-processing might be needed before the *pattern* could be plotted. In the prey-predator case, the models for each population produce the total number of individuals and the *pattern* is built out of those two pieces of information. We therefore need other components in our system that we call *artefacts* (see section 4.1) and which are pieces of information structuring our modeling environment.

To summerize, modeling consists in (1) detecting typical *patterns* characterizing the system, describing them in terms of *artefacts*' compositions and (2) building the sub-models that might best reproduce the phenomena explaining the emergence of the patterns. We structure the model with the same features that structure for an observer the studied system. Modeling and analyzing are based on those *patterns*. In a multi-modeling approach, experts assemble their models describing their inputs and outputs. Those inputs and outputs are influenced by *artefacts* building the *patterns* or produce them, enabling the experts to create the global model around the shared data.

3.2 Agent-based modeling

The agent paradigm enables different interacting entities to inhabit the same system without having any conditions on their internal mechanisms. Therefore, the multiagent approach is relevant to address multi-modeling. It enables the experts to choose different types of models and different approaches to best fit the requirements of their assumptions and available data, in other words the models which will best reproduce identified patterns. Second, multi-modeling enables experts to build separate simple models that can be incrementally tested and validated. The behavior of the global system is acquired through the observation of interacting entities, here the models. Hence, the simulation of complex systems is seen as a society of models in interaction [14]. The model is seen as a group of changeable components with no structural impact on other components (even though simulation outputs might be different). Coupling is made through the environment where the *patterns* are supported and not through events diffusion and synchronized state changes like in DEVS [15], used in the *Virtual Laboratory Environment* (VLE) [16]. The behavior of each agent influences or is influenced by the environment. Interactions are therefore indirect, through environment mediated influences and agents have their own execution time step and are not reactivated by external events for example. Furthermore, agents can locally observe, control or analyze the models. They can locally self-manage modeling processes: data scaling, observables creation, environment construction and modification through the creation of data in the environment.

4 *model-agent*-based solution

4.1 Concept of *artefact*

Patterns are composed by *artefacts* that we consider, in our system, as pieces of information produced by natural (human) or artificial agents. We therefore explicitly consider that any information in our modeling environment has been produced by a specific process (it is the case of data produced by observation for example). An *artefact* is the state in space and time of a specific variable and has one to n dimensions. In fact, *artefacts* are often multi-dimensional supporting spatialized variables. In ecology, *artefacts* are frequently maps. For example, we often focus on the temporal evolution of a fish population's spatial structure. This pattern would be composed in our approach by all the states of the *artefact* identifying the spatial structure of the population. The *artefact* would have two dimensions in this case (depth is not considered). Another *pattern* is the price of fish plotted versus the amount of fish catches. The *artefacts* involved in the creation of this *pattern* are the price's evolution in time and the amount of fish catches in time. *Artefacts* also support the interactions between (artificial and/or natural) agents. An agent has no impact on the system until it influences an *artefact* and therefore a *pattern*. In consequence, the *artefacts* (and the *patterns*) are the environments of the agents, which perceive, influence and are influenced by them. At last, *artefacts* and *patterns* support the analysis. Modelers watch the system through the *patterns* and might also observe the *artefacts*. Modelers center their observation on the very same structures that they believed were remarkable in the real system and whose behaviors they want to reproduce. Modelers start by building the environment of the agents, the structure of their model, and end up analyzing this environment only.

More precisely, an *artefact* handles data using a discrete n-dimensional matrix in a given space. Its specific semantic and shape are known by the agents. The shape is the definition of its span, defined in a given space. The *artefact* is defined on several or every axes of the space. Its span starts and ends somewhere on each axes. Therefore, applying a function to the *artefact* is formalized according to its spatial and temporal span and to its semantic. Modifying an *artefact* means applying geometrical or analytical transformations (projections, mean values, etc.) on its shape and scaling (logarithmic scale for instance) its semantic. At last, when an agent produces an *artefact*, it modifies its time stamp. Based on Lamport [17], this time stamp forbids any model using or writing an *artefact* to violate the causality principle. Hence, an *artefact* has the following description: $< S, f(S), A, t_{last} >$. S is the *artefact*'s semantic, $f(S)$ a function that for an element s of S associates $f(s)$ – it is the logarithm function for example and it enables to consider changing the scale of some data without loosing its semantic. t_{last} is the time stamp of the *artefact*. And, finally, A is a set of spatiotemporal axes on which the *artefact* is defined. A is never empty and contains at least the time axis' span. A span is described by: $< S, v_{init}, v_{final}, \delta v, unit >$. S is the semantic of the considered axis – t for instance would be the time. v_{init} is the initial position of the *artefact* on the axis. Hence, for the time, it would be the time of the *artefact*'s first execution step. In the same manner, v_{final} is the *artefact*'s final position on the axis and δv the smallest discretization step. It is, for the temporal span, the smallest update time step of the *artefact*. At last, $unit$ is the data's unit.

The formal description of a *pattern* is multi-scaled, just as a pattern itself. For instance, the *pattern* of the prey-predator populations and the *pattern* of the demographic structure of a population seem completely different and are expressed within different scales and dimensions. Therefore, we define a *pattern* as composed of (1) an *artefact*, (2) a set of *artefacts* or (3) the evolution in time of an *artefact* or a set of *artefacts*. For example, considering a population distribution in space, the corresponding *pattern* is composed of the states of the *artefact* that supports the distribution of the population at each step of time.

4.2 Agents' roles in modeling

The environment is, for an agent, all the *artefacts* with which it interacts. Therefore, an agent has input-output *features* that stand for the influences of the outside on the inner-system and those of the inner-system on the outside. Those *features* are considered in our system as active entities embodying (in the agent) the *artefacts* with which they are interacting. In other words, a *feature* modifies an *artefact* or is modified by it. But, more generally, as shown in figure 2, an agent manipulates a model and is composed by *features*. The model interacts with the *features*, several of which interact with *artefacts*. Hence, the agent controls the interaction with the environment, while its internal model can be executed without knowing anything about the environment. The agent's behavior or its capacity to process the modifications of its internal variables or of the environment (*artefacts*), depends on the modeling approach chose for its internal model. But in any case, the agent's goal is to produce *artefacts*. The *features* are variables, parameters and constants and the agent can : (1) maintain local control on variables of its model or (2) use specific heuristics, to be stated, to initialize its model's parameters.

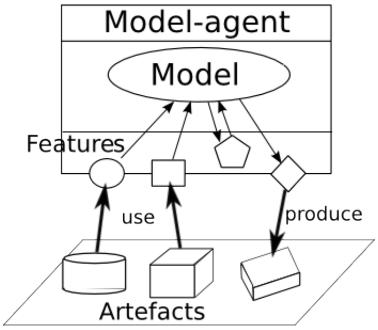


Fig. 2 Agents, features and *artefacts*: an agent interacts with the *artefacts* via features.

Three roles of agents in modeling were identified: (1) a *model-agent* handles an expert model and produces *artefacts*; (2) an *observer-agent* watches *artefacts* or *patterns* and produces observable entities (images, 3D objects, etc.); and finally, (3) a *controler-agent* watches *artefacts*, compares them to specific values – described by experts, by preprocessed scenarios, by real data observations or by other *artefacts*– and triggers specific behaviors if the *artefacts* reach unwanted values (recording, sending messages to modelers, executing heuristics).

Model-agents can manipulate four types of models (see figure 3): (1) models based on differential equation systems, that we name *intensional models*, because only the properties of the system are specified, not the components; (2) individual-based models, that we call *emergent models*, because the behavior of the system is not given but is the result of its elements' behaviors and interactions; (3) participatory models, played by human experts; (4) models built out of data series (temporal series that can be spatialized), that we will designate as *extensional models*, because each and every state of the system is described, not its behaviors. Therefore, intensional *model-agents* are structured by *features*, parameters or variables of an equation system. With those equations, intensional *model-agents* process their state at each step of time and modify the values of the out-

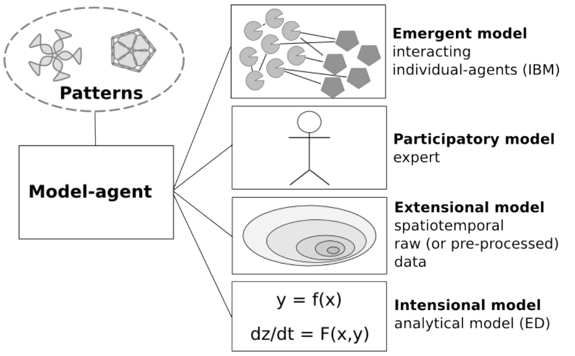


Fig. 3 Pattern-oriented multi-*model-agents* simulation: on the right, the different types of models a *model-agent* can handle.

put *features*. Emergent *model-agents* are composed of multiagent systems using input *features* and modifying output *features*. At each step of time, emergent *model-agents* make their multiagent systems evolve of one or more cycle of an internal frequency that depends on the phenomenon they try to reproduce [18]. Participatory *model-agents* enable a human actor to participate to simulations. Participatory *model-agents* are composed by *features* that enable humans to interact with the system through *artefacts*' influences or observations (with proper visualization tools). At each time step, humans analyze their *features* and modify them. At last, extensional *model-agents* are not influenced by outer system. At each step of time, they follow their evolution scenarios prepared for their *features*.

At last, we need one last set of components in order to give agents the capacities to modify their environment. We therefore specify *operations* as pre-defined functions that can be applied on *artefacts* in order to change their shapes. Because agents can perceive the semantic and shape of *artefacts*, they can manipulate *operations* in order to modify those *artefacts*. Depending on the *artefact* it has to read or produce, an agent can manipulate one or several *operations* to transform the *artefact*. For example, if an agent needs a specific *artefact*, it first looks in the environment for an *artefact* with the proper semantic and the closest shape. If no such *artefact* is found, the agent tells the modelers about its incapacity to function. Otherwise, it uses *operations* executing the proper transformations to get the *artefact* with the required shape. Modelers are explicitly told about this process and the *operations* are made visible in the environment. Therefore, agents can locally build the global model and tell modelers about it or about their incapacity to function if needed. Of course, using *operations* on data means modifying this data. But because modelers create the *operations* and because the automatic use of *operations* is explicit in the model (or can be decided by the modelers), modelers can control their proper use.

4.3 Interaction and synchronisation

A *feature* can be either influenced by an *artefact*, or can influence one. This influence is not linked to the agent's activity. We will in consequence, regarding an agent, speak of *input features* and *output features*. It is therefore necessary to order the activation of the different entities as follows: (1) activation of *input features* (updating the internal representations of the agents) ; (2) agents' activation, its internal model is executed ; (3) activation of the *output features* (updating the *artefacts*). Hence, in the global organization of our multi-model system, agents, and specifically *model-agents*, are synchronized with the environment. The *features*' activities are not linked with those of the agents. An agent may therefore change and evolve without its *features* having any activities. There would in that case be no modifications of its *input features*. Agents are highly detached from the rest of the system and the system does not lie on any specific agent. Hence, it is easy to add or remove dynamically new or existing agents.

Finally, *operations* are a type of *features*. They are simply *features* that modify the data they manipulate. Such *features* exhibit the shape requested by the agent while reading or producing an *artefact* with a different shape. The reading and writing processes are just not simple reading and writing but execute the functions specified by the *operations*.

5 Application to exploited ecosystems

Description of the studied system

In this example, we want to understand the processes involved in the dynamics of a theoretical fish population. This population is located in its habitat in the Bay of Biscay and exploited by a fishery. We observe the shift of the fish demographic distribution to the North of the Bay and we want to understand which phenomena are the causes of this migration. Hence, the *pattern* we focus on and we want to reproduce is this shift of the population distribution from the South of the Bay to the North. Furthermore, we notice that the population's total biomass does not grow during the process and that it is more or less stable in time. At last, it is likely that the fish population is affected by global warming, which increases the temperature of the Bay from South to North progressively. This system is of course theoretical and illustrates our modeling methodology and simulation tool. But, we believe the system is instructive enough to exhibit what our approach and simulator can do now and what should be done to improve it.

First simulation scenario: reproducing the *pattern*

We first select the processes that will best explain the *pattern* we have identified. We add to the global model the fish population model and the global warming phenomenon in order to broadly explain the *pattern*. To model the impact of the global warming on the fish population, we describe the habitat of the fish as a function of the latitude, the depth and the temperature in the Bay in a specific *model-agent* that produces the habitat in terms of an affinity map. Therefore, we consider the habitat as an *artefact* shared by both the fish population and habitat models. We do not want to add the *model-agent* manipulating the model of the fishing activity yet, but we want the *model-agent* of the

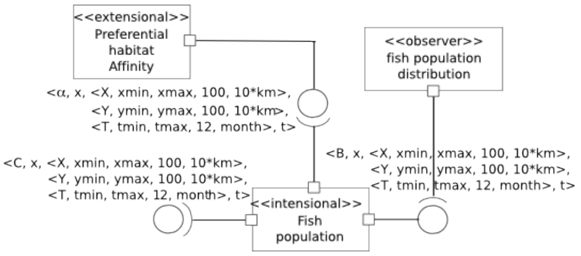


Fig. 4 First simulation scenario – an intensional *model-agent* describes the fish population and all *artefacts* are formally described (see section 4.1).

fish population to be able to integrate catches, therefore we add an *artefact* supporting the catches, even though no *model-agent* will produce it for now. This system is described in figure 4. We therefore have three *model-agents* in the global model: (1) the extensional *model-agent* producing the habitat distribution, (2) the intensional *model-agent* for the fish population and (3) an *observer-agent* creating the *pattern* we focus on and we want to observe. There are also three *artefacts*: (1) for the affinity distribution (α), (2) the fish biomass distribution (B) and (3) the fish catches (C – which is initialized to zero – no fishing). Each *artefact*'s semantic and shape is detailled. For instance, if we consider the *artefact* of the fish distribution, its description is $< B, x, < X, x_{min}, x_{max}, 100, 10 * km >, < Y, y_{min}, y_{max}, 100, 10 * km >, < T, t_{min}, t_{max}, 12, month >, t >$. The *artefact*'s semantic is B standing for biomass. This biomass is described directly (x). Then, three axes are listed as the biomass is expressed on the X, Y and T axes. It is spatially (X and Y) and temporally (T) distributed. If we consider the span of B on one axis, X for instance, we can see that this span starts at x_{min}, ends at x_{max}, has a discretization step of 100, and that its unit is $10 * km$. Finally, the last item t is the time stamp.

We have stated a system of differential equations modeling fish populations in trophic interactions. We do not detail it here, but we use part of it to model the fish population of our example. We only use one equation (for one population):

$$\begin{aligned} f(x,y,t+1) &= f(x,y,t) + \delta t \frac{\partial f}{\partial t}(x,y,t) \\ &\quad -C(x,y) \times f(x,y,t) \end{aligned} \tag{1}$$

$$\begin{aligned} \frac{\partial f}{\partial t}(x,y,t) &= (\eta \nabla f(x,y,t) - \nabla a(x,y,t)).\nabla f(x,y,t) \\ &\quad +(\eta \Delta f(x,y,t) - \Delta a(x,y,t)).f(x,y,t) \end{aligned} \tag{2}$$

We have removed the parts that model the trophic interactions and only kept those regarding the diffusion (density-dependence) and advection (tropism towards affinity) processes. Furthermore, the parameter representing the natural growth of the fish population is also set to zero and not exhibited here. Therefore, we do not control artificially the growth of the population. The fish population is modeled using the equation 5. At $t+1$, the number of fish in (x, y) is processed using the number of fish in (x, y) at t, $\frac{\partial f}{\partial t}(x, y, t)$ and the quantity of fish captured at t. The equation 3 exhibits the parts describing the diffusion-advection processes.

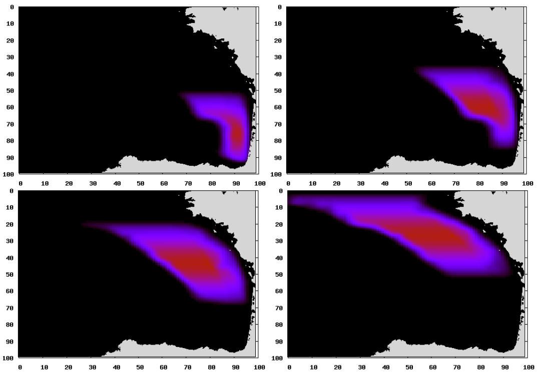


Fig. 5 The fish preferential habitat (in terms of affinity distribution) influenced by global warming at time steps 100, 1000, 2000 and 3000 – output of an extensional *model-agent*.

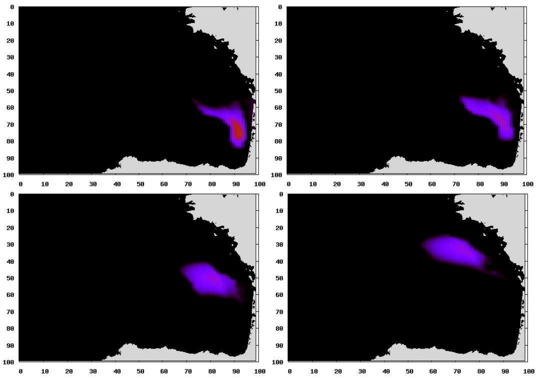


Fig. 6 First simulation scenario: the distribution of the fish population at time steps 100, 1000, 2000 and 3000 – output of an intensional *model-agent*.

Figure 5 shows the output of the extensional *model-agent* producing the habitat of the fish population (an *observer-agent* was used to produce those figures, but it does not appear in the diagram) and figure 6 shows the output produced by the *model-agent* of the fish population model.

Second simulation scenario: adding an agent without impact on the system

Having produced the *pattern*, we now add the model of the fishing activity to the global model. It is made easy with our approach, as we just have to add the corresponding *model-agent*. The fishing vessels are modeled with an emergent model. Fishermen are distributed in two regions of the Bay (North and South) and are structured by two processes: (1) a perception of the two regions enabling the fishermen to choose the region with the larger stock ; (2) a fishing activity – the fishermen, located in a region (North or South), choose the best areas to fish and produce catches. But, in order to have a more interesting system, we chose to have the fishing activity model expressed in another time scale than

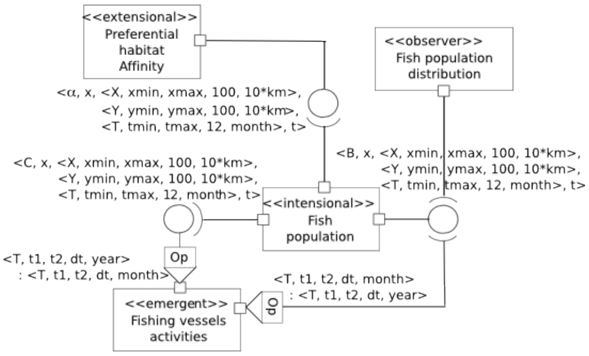


Fig. 7 Second simulation scenario: the emergent *model-agent* of the fishing activity is added to the global model.

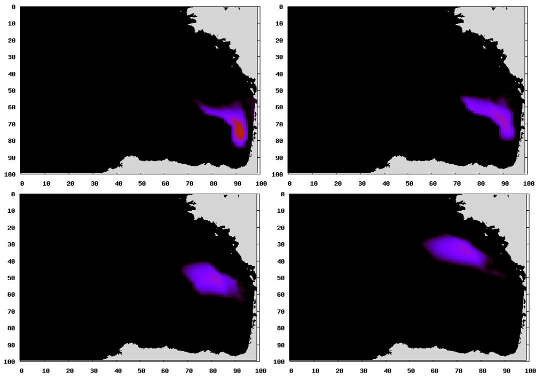


Fig. 8 Second simulation scenario: the distribution of the exploited fish population at time steps 100, 1000, 2000 and 3000 – output of an intensional *model-agent*.

the one of the fish population model. While the *artefact* of the fish distribution is produced monthly and the *model-agent* of the fish population is influenced by an *artefact* of catches expressed monthly, we modeled the fishing activity with a period of time of a year. Therefore, the *model-agent* of the fishing activity has to transform its input and output *artefacts* in order to function properly. Hopefully, we had added the two following *operations* in the system: (1) the first one reads (inputs) every month (simulation time) a two-dimensional *artefact* and sets its value to a two-dimensional matrix which is the mean value of the twelve last inputs; (2) the second one, being written by the *model-agent* every year, writes every month the twelfth of its value to the *artefact*. To summarize, the *model-agent* of the fishing activity, when added to the global model, detects that the *artefacts* it needs to read and produce do not have the proper shapes. Therefore, it uses the proper *operations*, just like normal *features*, to change the *artefacts*' shapes. Figure 7 shows the resulting model and figure 8 shows the resulting output of the fish population *model-agent*. Unsurprisingly and therefore not shown here, the fishermen move progressively to the North of the Bay when the stock gets bigger there as the fish population migrates to the North.

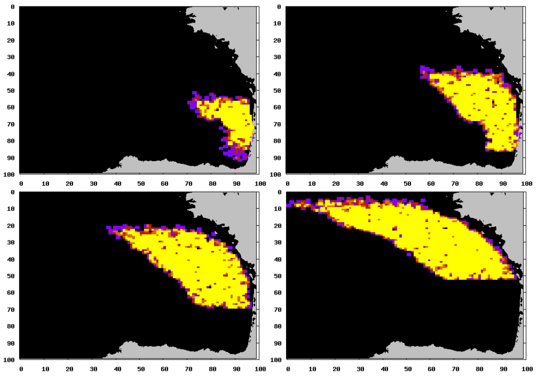


Fig. 9 Third simulation scenario: the distribution of the exploited fish population at time steps 100, 1000, 2000 and 3000 – output of an emergent *model-agent*.

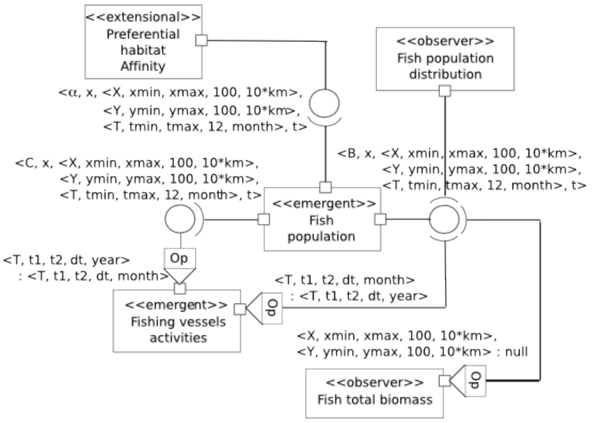


Fig. 10 Fourth simulation scenario: An *observer-agent* is added to the global model.

Third simulation scenario: locally desegregating a *model-agent*

We now change the model of the *model-agent* of the fish population. It enables us to have a better control on the distribution-advection processes avoiding at the same time the use a global growth parameter. It especially enables us to exhibit the local change of a model with no impact on the structure of the global model. The individual-based model is described as follows: each individual perceives its local environment and the individual processes modeled are (1) a density-dependence that kills the individual if the local density is superior to a specific maximum density; (2) a spatialized reproduction – an individual reproduces itself depending on the affinity to its local environment and on its size; (3) a classical growth in weight and size, based on Von Bertallanfy; (4) a mortality based on the age of the individual (longevity); at last, (5) a motion process that creates new individuals somewhere in their preferential habitat. The resulting output of the emergent *model-agent* are exhibited in figure 9.

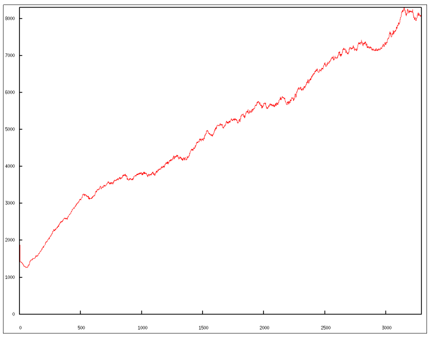


Fig. 11 Fourth simulation scenario: the total biomass of the fish population during the simulation – output of an *observer-agent*.

Fourth simulation scenario

We now want to observe the evolution of the global biomass of the fish population. We add a new *observer-agent* to the system. Its specified goal is to observe an *artefact* with the semantic B and defined on the sole time axis (T). The only *artefact* with the corresponding semantic is defined on the axes X and Y (produced by the fish population *model-agent*) (and the T axis of course). This map of the fish population in space is two-dimensional (latitude and longitude). Therefore, the *observer-agent* uses *operations* to read the *artefact*. Simple projections of the initial *artefact* are done in order to transform the 2D *artefact* into 0D value. The system is described in figure 10) and the output of the *observer-agent* plotting the population's biomass is shown in figure 11.

The fish population increases in time (figure 11), when we wanted it to be stable. Obviously, this growth is due to the increase in size of the habitat, giving the fish more "room" and decreasing the local density. Our inability to keep the population biomass stable could be obviously explained by our modeling of the habitat, which might be not relevant. The modeling of the fishing activity could also be a reason as it was made constant in time. An increase in time of the fishing activity could explain the stable biomass of the fish population. Of course, this example is theoretical and we chose the initial assumptions in order to build a didactic simulation scenario, exhibit our methodology and show how some automation could be envisioned.

6 Conclusion and perspectives

We exhibit through our simulation example of an exploited fish population our modeling methodology. We tested several key aspects of our system – locally adding agents without impact on the global model, changing the model of *model-agents* or having agents locally control or change their environment. Our *model-agent*-based solution enables experts to incrementally build complex models of complex systems. The modeling is centered on the data and especially on identified *patterns* that *model-agents* have to reproduce. The environment, composed by *patterns* and *artefacts*, both being formally described, enable experts to build their model around the structures that best describe the studied system. The formal description of the *artefacts* also enables agents to automatically manipulate them and change their shapes. At last, agents, conceived as black boxes, which behaviors are structured by the *artefacts* they have to produce, enable us to consider the global model as a system of models.

However, our proposition needs more work in order to exhibit more relevant automatic behaviors and simulate even more complex scenarios. Our example validates several key points, but we are working on having real collaboration between agents. The work on *operations* has to be carried on, and of course, more complex operations have to be added to the system. In fact, we now especially focus on: (1) describing the society of models in terms of rules on the global model; (2) the agents in the simulation process – agents could execute automatic simulation scenarios (sensibility analysis) for example. On the long term, the self-organization concept introduced by the agent paradigm means that agents could self-measure part of their models, self-transform themselves or have better adaptation behaviors (like self-calibration for instance).

This work was carried out with the financial support of Chaloupe, a French National Agency for Research funded project.

7 References

[1] Volker Grimm, Eloy Revilla, Uta Berger, Florian Jeltsch, Wolf M. Mooij, Steven F. Railsback, Hans-Hermann Thulke, Jacob Weiner, Thorsten Wiegand, and Donald L. DeAngelis. Pattern-oriented modeling of agent-based complex systems: lessons from ecology. *Science*, 310:987–991, 2005.

[2] Thorsten Wiegand, Florian Jeltsch, Ilkka Hanski, and Volker Grimm. Using pattern-oriented modeling for revealing hidden information: a key for reconciling ecological theory and application. *OIKOS*, 100:209–222, 2003.

[3] R. Duboz, E. Ramat, and P. Preux. Towards a coupling of continuous and discrete formalisms in ecological modelling - influences of the choice of algorithms and results. In N. Giambiasi and C. Frydman, editors, *Proc. 13th European Simulation Symposium*, pages 481–487, October 2001.

[4] D.L. DeAngelis, K.A. Rose, and M.A. Huston. Individual-oriented approaches to modeling ecological populations and communities. In Levin S.A., editor, *Frontiers in mathematical biology*, pages 390–410. Springer Verlag, 1994.

[5] Volker Grimm. Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological Modelling*, 115:129–148, 1999.

[6] J. Hjort. Fluctuation in the great fisheries of northern europe reviewed in the light of biological research. *Rapports et Procs-Verbaux des Runions*

du Conseil International pour l'Exploration de la Mer, 20:1–28, 1914.

[7] D.L. DeAngelis, B.J. Shuter, M.S. Ridgeway, P. Blanchfield, T. Friesen, and G.E. Morgan. Modeling early life-history stages of smallmouth bass in ontario lakes. *Transaction of the American Fisheries Society*, -, 1991.

[8] A. Drogoul and J. Ferber. Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In *Decentralized A.I.*, North-Holland, 1992. MAAMAW'92, Elsevier.

[9] Steve Railsback, Bret Harvey, Steve Jackson, Roland Lamberson, and Walt Duffy. California individual-based fish simulation system, trout instream flow model formulation. Technical report, Humboldt State University, Arcata, California, August 1999.

[10] Leigh Tesfatsion. Agent-based computational economics. *ISU Economics working paper*, 1, August 24 2003.

[11] B. LeBaron. A builder's guide to agent based financial markets. *Quantitative Finance*, 1(2):254–261, 2001.

[12] Steven F. Railsback. Concepts from complex adaptive systems as a framework for individual-based modelling. *Ecological Modelling*, 139:47–62, 2001.

[13] Volker Grimm and Steven F. Railsback. *Individual-based Modeling and Ecology*. Princeton University Press, 2005.

[14] Jacques Tisseau. Virtual reality – in virtuo autonomy. Accreditation to Direct Research, University of Rennes 1, 2001.

[15] B. Zeigler. *Theory of Modeling and Simulation*. Wiley Interscience, New York, 1976.

[16] E. Ramat and P. Preux. Virtual laboratory environment (VLE): A software environment oriented agent and object for modeling and simulation of complex systems. *Journal of Simulation Practice and Theory*, 11:45–55, mars 2003.

[17] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21:558 – 565, 1978.

[18] P. Redou, S. Kerdelo, C. Le Gal, G. Querrec, V. Rodin, J.-F. Abgrall, and J. Tisseau. Reaction-agents: First mathematical validation of a multi-agent system for dynamical biochemical kinetics. *Lecture Notes in Computer Science (Progress in Artificial Intelligence)*, 3808:156–166, 2005.