

УДК: 004.9+519.6

## Об одном универсальном методе построения моделей для сложных многоагентных систем

А. А. Стеряков

Самарский государственный аэрокосмический университет имени академика С. П. Королёва  
(национальный исследовательский университет),  
Россия, 443086, г. Самара, Московское шоссе, д. 34

E-mail: steryakov@mail.ru

Получено 18 апреля 2013 г.,  
после доработки 3 июля 2013 г.

В работе предлагается универсальный метод построения агентных имитационных моделей сложных систем, предполагающий их компьютерную реализацию на языках объектно-ориентированного программирования. Метод определяет способ построения математических моделей агентов и их взаимодействия, а также описывает архитектуру комплекса программ для имитации динамики моделируемой системы. Эффективность предлагаемого метода иллюстрируется примерами его применения для моделирования сложных систем из двух областей: экономической (модель финансового рынка с неоднородными агентами) и биологической (пространственно-временная имитация взаимодействия биологических популяций).

Ключевые слова: агентное моделирование, методология моделирования, мультиагентные системы, объектно-ориентированная модель, объектно ориентированное программирование, сложные системы

### A universal method for constructing the simulation model of complex multi-agent systems

A. A. Steryakov

*Samara State Aerospace University named after academician S. P. Korolyov (National Research University),  
34 Moskovskoye Shosse, Samara, 443086, Russia*

**Abstract.** — This paper presents a universal method for constructing an agent-based model of complex systems for their further clear computer representation by means of object-oriented programming languages. The method specifies both steps of model developing from the mathematical description of the system to the determined architecture of the program simulating the system. The efficiency of the method is illustrated by the construction of the two simulation models for the complex systems of various origins: the interactive simulation of the stock exchange and space-time simulation of biological species competition.

Keywords: agent-based model, complex systems, modeling methods, multi-agent systems, object-oriented model, object-oriented programming

Citation: *Computer Research and Modeling*, 2013, vol. 5, no. 4, pp. 513–523 (Russian).

Работа выполнена при частичной поддержке РФФИ (проект № 13-01-00790-а).

## Введение

Агентное моделирование (АМ) как один из наиболее актуальных способов создания имитационных моделей в настоящее время является интенсивно развивающейся областью как за рубежом, так и в отечественной науке [Macal, North, 2005; Castle, Crooks, 2006; Борщёв, 2004; Замятина, Чудинов, 2010; Городецкий, 2012b]. Связано это с тем, что при решении задачи моделирования современных систем исследователь сталкивается с проблемой описания сложной структуры взаимодействующих элементов, которые, кроме того, сильно неоднородны и не подлежат типизации по своим свойствам. Аналитические методы и традиционное дискретно-событийное моделирование (равно как и другие часто применяемые методы: системная динамика, динамические системы [Борщёв, 2004]) даже в случае возможности их применения оказываются неэффективными. Например, в традиционных моделях такой прикладной области как моделирование экономических рынков [Chiarella, He, 2001; Brock, Hommes, 1998] имеют место допущения о рациональности рынка, однородных посредниках, долговременном равновесии и других искусственных упрощениях, которые позволяют решать задачу аналитически. Отказ от таких предположений приводит к невозможности использования прежних методов, которые к тому же имели и такие недостатки, как сложность добавления деталей к уже построенным моделям, их модификации и расширения. В силу вышесказанного прибегают к применению агентных моделей, выстраиваемых снизу вверх, то есть так, что глобальная динамика системы формируется за счет взаимодействия автономных агентов. Гибкость и адекватность АМ повышается при введении когнитивных агентов, способных к адаптации. Еще одним катализатором развития АМ служит быстро увеличивающаяся вычислительная мощность ЭВМ, допускающая моделирование огромного числа независимых объектов даже на ПК, имеющемся в распоряжении каждого отдельного исследователя.

Успешному применению АМ на практике мешает ряд проблем, связанных в первую очередь с инструментарием для разработки таких моделей. Для того чтобы выполнить моделирование агентного типа, необходимо, во-первых, описать все элементы системы, во-вторых, задать способ их взаимодействия. Сделать это можно различным образом [Nikolai, Madey, 2009], и практически каждый исследователь подходит к решению этой задачи, используя собственные разработки или немногочисленные пока программные пакеты для проведения агентного моделирования [Politopoulos, 2007]. Существующие системы, как правило, достаточно трудны для овладения ими (Repast, Netlogo [Macal, North, 2005]). Кроме того, изучив новый синтаксис и методы разработки, зачастую можно столкнуться с ситуацией, когда инструмент не подходит для конкретной модели или не предоставляет достаточной гибкости для проведения исследования. Отмечают также сложности, связанные с невозможностью полного представления моделей [Castle, Crooks, 2006]. Другие разработки в этом направлении (AnyLogic [Карпов, 2005; Сидоренко, Красносельский, 2009]) представляют собой дорогостоящие коммерческие программные пакеты, ориентированные на использование на предприятиях и не всегда удобные для использования в исследовательских целях.

В настоящее время актуальна задача универсализации методов и средств разработки агентных моделей [Городецкий, 2012b; Городецкий, 2012a]. Прежде всего, необходимо выработать методологический подход в моделировании такого типа, предоставить исследователю некоторый общий алгоритм действий, описывающий весь цикл построения АМ — от формулировки задачи и ее математической формализации до реализации комплекса программ для проведения численных экспериментов. В случае выполнения этого значимого условия развитие данной ветви моделирования ускорится и выйдет на уровень традиционных методов. Особенно важно предложить средства моделирования, которые будут инвариантными относительно сфер применения и сложности описываемых систем.

В работе предлагается достаточно универсальный метод построения моделей для мульти-агентных систем и последующей их компьютерной реализации. В первой части статьи описаны подход к моделированию и архитектура, позволяющая унифицировать описание систем из различных предметных областей. Далее описан подход к реализации комплекса программ для проведения численного моделирования исследуемых систем. В следующих частях приводится описание созданных с применением предлагаемой методики моделей [Стеряков, 2012a; Стеряков, 2012b] реальных систем из двух различных областей приложения. В заключении излагаются основные выводы и результаты работы.

## Объектно-ориентированные модели. Универсальная архитектура

Существование ныне хорошо развитой концепции объектно ориентированного программирования (ООП), во-первых, позволяет использовать ООП для компьютерной реализации агентного моделирования, а, во-вторых, наталкивает на мысль использовать некоторые принципы этой концепции для построения самих моделей. А именно, ориентируясь на последующую реализацию, классифицировать все элементы системы и далее, следуя принципу инкапсуляции, моделировать элементы системы в виде уединенных объектов, наделенных свойствами и поведением (то есть вектором состояния элемента и функциями, описывающими реакцию элемента на текущее состояние системы).

Таким образом, будем говорить об *объектно ориентированных моделях* (далее ООМ), понимая под этим модели, относящиеся к классу мультиагентных имитационных моделей и построенные по принципам, которые естественным образом обеспечивают их дальнейшую реализацию в виде комплекса программ на языках ООП.

Автором предлагается метод построения таких моделей, основные этапы которого, связанные с анализом элементов системы, могут быть сформулированы следующим образом:

1. Классификация элементов системы.
2. Описание всех характеристик каждого класса элемента в виде вектора состояния и всех реакций элемента в виде функций.
3. Выделение функций, отвечающих за автономное изменение состояния элемента.
4. Выделение функций, связанных с взаимодействием элемента с окружающей средой.
5. Формирование минимального интерфейса для взаимодействия путем выделения параметров, содержащих информацию, достаточную для описания внешних взаимодействий агента.

Предлагается определенная архитектура ООМ, обладающая достаточной универсальностью, которая включает в себя три основных понятия: агент – заявка – стекковое взаимодействие.

**Агент.** В системе, макросостояние которой характеризуется вектором  $\bar{s}(t)$ , для каждого класса элементов строится модель  $i$ -го агента, характеризующаяся, как было сказано выше, вектором состояния  $\bar{v}^i(t) = (v_1^i(t), v_2^i(t), \dots, v_n^i(t))$  и функциями двух назначений:

1. Первая (в дальнейшем — *функция преобразования* состояния агента) отвечает за независимое от других элементов системы изменение состояния агента, то есть вектора состояния (или его части), на каждом временном шаге. Данная функция определяет автономность агента

$$\bar{v}^i(t + \Delta t) = g^i(v_1^i(t), v_2^i(t), \dots, v_m^i(t), \bar{s}(t)), m \leq n$$

2. Вторая (в дальнейшем — *поведенческая функция* агента) формирует вектор параметров  $\bar{b}^i = (b_1^i, b_2^i, \dots, b_m^i)$ , который будем называть заявкой и который используется для осуществления взаимодействия агентов в системе. Таким образом, поведенческая функция определяет реакцию элемента системы на внешнюю среду

$$\bar{b}^i = \phi^i(\bar{v}^i(t), \bar{s}(t)).$$

Необходимо отметить, что предлагаемая архитектура не накладывает никаких ограничений на поведенческую функцию и функцию преобразования. В данном случае функция понимается в широком смысле, то есть может быть задана аналитически или представлена любым алгоритмом преобразования входных данных в выходные, например, нейросетью. Это особенно важно при реализации когнитивных агентов, способных к адаптации.

**Заявкой** называется вектор параметров  $\bar{b}^i$ , определяемый поведенческой функцией, который задает протокол информационного обмена между агентом и системой взаимодействия. Таким образом, заявка служит средством внутренней коммуникации в системе, тем самым обеспечивая функцию передачи сообщений.

Заявка должна содержать всю необходимую информацию для того, чтобы могло быть осуществлено взаимодействие.

**Стековое взаимодействие.** Взаимодействие агентов в предлагаемой архитектуре происходит итеративно и реализуется с помощью стека. В данном случае под стеком подразумевается упорядоченный набор заявок:

$$L = \{\bar{b}^1, \bar{b}^2, \dots, \bar{b}^n\}.$$

Кроме того, для каждой отдельной задачи может быть использован набор стеков, в которые по определенным критериям будут распределяться заявки при сборе. В общем случае взаимодействие осуществляется в три этапа:

1. Сбор заявок от каждого агента в стек  $L = \{\bar{b}^i = \phi^i(\bar{v}^i(t), \bar{s}(t))\}$ .
2. Обработка стека, то есть осуществление взаимодействия  $\hat{L} = G(L)$ .

Взаимодействие осуществляется путем удовлетворения заявок. При рассмотрении каждой заявки производится анализ представленной в ней информации, исходя из правил, определенных предметной областью моделируемой системы. На основании результатов этого анализа происходит поиск других необходимых заявок для взаимного удовлетворения. Перебор стека может проходить в несколько итераций до выполнения условий удовлетворения всех возможных заявок.

3. Соответствующее изменение векторов микро- и макро- состояний системы:

$$\begin{cases} \bar{v}^i(t + \Delta t) = \bar{f}^i(\{\bar{v}(t)\}, \{\bar{b}\}, \bar{s}(t), \bar{s}(t + \Delta t)), \\ \bar{s}(t + \Delta t) = \bar{F}(\{\bar{v}(t + \Delta t)\}, \bar{s}(t)). \end{cases}$$

Предлагаемое стековое взаимодействие позволяет учесть больше связей между объектами, чем при их последовательном переборе.

## Метод реализации ООМ на языках ООП

Построенная предложенным методом ООМ, как уже было отмечено, может логичным образом быть представлена в виде программы, написанной на языке объектно ориентированного программирования. Для этого необходимо реализовать структуру классов, UML-диаграмма<sup>1</sup> которой представлена на рисунке 1.

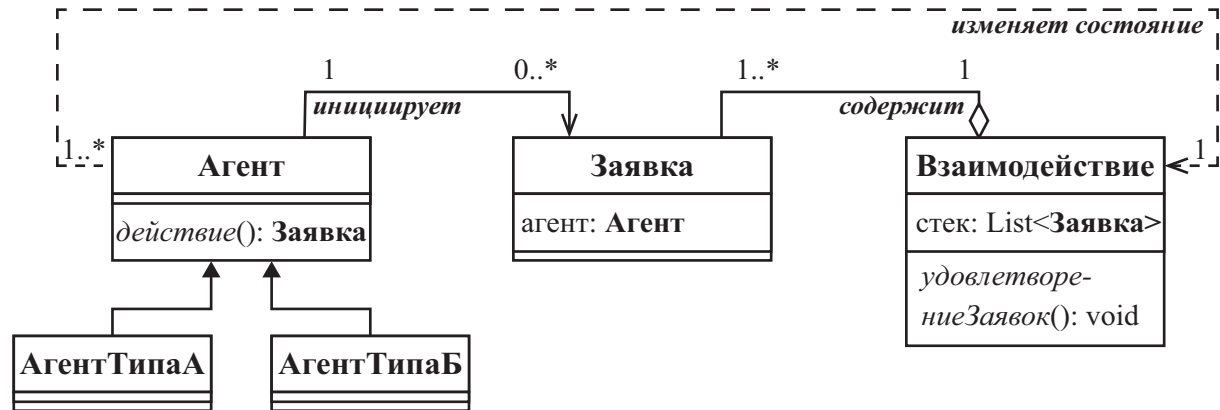


Рис. 1. Структура классов для реализации ООМ

Модели агентов описываются иерархией классов, являющихся наследниками единого интерфейса или абстрактного класса (на диаграмме «Агент»), общего для всех агентов и декларирующего единственный обязательный элемент – метод, отвечающий за поведенческую функцию агента и выдающий на выходе заявку (экземпляр класса «Заявка»). Остальные классы соответствуют отдельным типам агентов, отличающихся между собой поведением или набором внутренних параметров.

Класс «Заявка» представляет собой класс данных, состоящий в простейшем случае из фиксированного набора параметров и ссылки на объект-родитель (на соответствующий экземпляр класса агента). Все поля инициализируются при создании каждого экземпляра этого класса.

Осуществление взаимодействия делегируется отдельному классу («Взаимодействие»), который среди полей в обязательном порядке имеет нефиксированный в размере стек (или систему стеков) заявок, куда каждую итерацию будут поступать заявки от агентов. Также данный класс характеризуется методом, отвечающим за удовлетворение заявок. В этом же методе посредством ссылок, хранящихся в объектах класса заявки, реализуется алгоритм изменения векторов состояния каждого агента.

В зависимости от конкретных целей исследования в программе могут быть описаны другие дополняющие элементы, но в целом описанной структуры достаточно для реализации ООМ с предложенной архитектурой.

Чтобы начать симуляцию с помощью созданного комплекса, в исполняемом методе необходимо создать и инициализировать коллекцию агентов с необходимым количеством различных типов моделируемых объектов, а также экземпляр класса взаимодействия. Далее в цикле с нужным числом итераций у каждого агента вызывается отвечающий за действие метод, в который передается ссылка на экземпляр класса взаимодействия, а затем вызывается метод класса взаимодействия для удовлетворения всех заявок.

<sup>1</sup> UML (англ. Unified Modeling Language – унифицированный язык моделирования) – язык графического описания для объектного моделирования в области разработки программного обеспечения.

## ООМ для финансовой системы

В качестве иллюстрации применения метода создания ООМ и описанной архитектуры рассмотрим модель из области финансов, а именно модель биржи с одним типом акций [Steryakov, 2012; Стеряков, 2011].

Цели создания модели заключаются в получении временных рядов, статистические характеристики которых схожи с реальными аналогами, а также в предоставлении инструмента для проведения исследования микро- и макро- динамики рынка. По сути поставленных задач модель является продолжением деятельности Леви–Леви–Соломона [Levy et al., 2000; Levy et al., 1994], но отличается структурой и компьютерной реализацией. Также в рамках данной модели решены другие задачи, не рассматриваемые этими авторами ранее, в частности генерация временных рядов с характерными статистическими характеристиками [Стеряков, 2012a; Cont, 2001].

Согласно предложенной архитектуре для ООМ необходимо описать три основных составляющих. Вектор состояния агента имеет в данной модели следующий вид:

$$\bar{v}^i = \{w^i(t), \pi^i(t), \mu^i(t), \sigma^i(t)\},$$

где  $w^i(t)$  — общая стоимость рискового и безрискового активов в портфеле  $i$ -го агента,  $\pi^i(t)$  — доля стоимости рискового актива,  $\mu^i(t)$ ,  $\sigma^i(t)^2$  — соответственно математическое ожидание и дисперсия в прогнозе изменения стоимости рискового актива. Такой вид вектора определяется математической моделью агента, предложенной автором и более подробно описанной в [Стеряков, 2012a].

Функция преобразования состояния агента в данном случае определяет изменение  $\mu^i(t)$  и  $\sigma^i(t)^2$ , которое в частном случае можно записать как

$$\begin{cases} \mu^i(t + \Delta t) = \lambda^i \mu^i(t) + (1 - \lambda^i) \rho(t), \\ \sigma^i(t + \Delta t)^2 = \lambda^i \sigma^i(t)^2 + (1 - \lambda^i) (\rho(t) - \mu^i(t))^2, \end{cases}$$

где  $\rho(t)$  — относительное изменение цены,  $\lambda^i \in (0; 1)$  — параметр, определяющий степень влияния последнего изменения цены на прогноз.

Далее в результате максимизации функции полезности можно аналитически получить [Стеряков, 2012a] выражение для «наиболее выгодного» агенту соотношения между рисковым и безрисковым активами на следующем шаге, а значит, и для соответствующего количества акций:

$$\frac{w^i(t)}{P(t)} \hat{\pi}^i(t) = \frac{w^i(t)}{P(t)} \frac{\mu^i(t + \Delta t) + \frac{\delta}{P(t)} - r}{\beta w^i(t) \sigma^i(t + \Delta t)^2} = \frac{\mu^i(t + \Delta t) + \frac{\delta}{P(t)} - r}{\beta P(t) \sigma^i(t + \Delta t)^2}.$$

Таким образом, поведенческая функция в данном случае, исходя из состояния агента, то есть его прогноза, и текущего вектора макросостояния системы, выдает вектор заявки:

$$\bar{b}^i = \{d^i, n^i, p^i\},$$

состоящий из количества акций  $n^i = \frac{w^i(t)}{P(t)} (\hat{\pi}^i(t) - \pi^i(t))$ , направления торговли  $d^i = \text{sign}(n^i)$  и цены  $p^i$ , определяемой случайным отклонением от текущей:

$$p^i(t + \Delta t) = \begin{cases} P(t)(1 - \nu\xi), & d^i > 0, \\ P(t)(1 + \nu\xi), & d^i < 0, \end{cases}$$

где  $\xi \sim U[0, 1)$  — равномерно распределенная случайная величина.

Модель взаимодействия описывает несколько этапов: 1) заполнение стека, а именно двух упорядоченных стеков с заявками на покупку и на продажу, 2) операцию переформирования стека (для моделирования процесса взаимных уступок на рынке) и 3) нетривиальный в плане формализации процесс удовлетворения всех заявок с соответствующим изменением векторов состояния всех агентов и вектора состояния всей системы.

В результате численного моделирования в программе, реализованной по данной объектно-ориентированной модели, была воспроизведена динамика системы, представляющая самоподдерживающийся процесс торговли. А именно в закрытой системе устанавливается равновесная цена, зависящая только от внутренних параметров системы, и вокруг этой цены происходят хаотические колебания (рисунок 2).

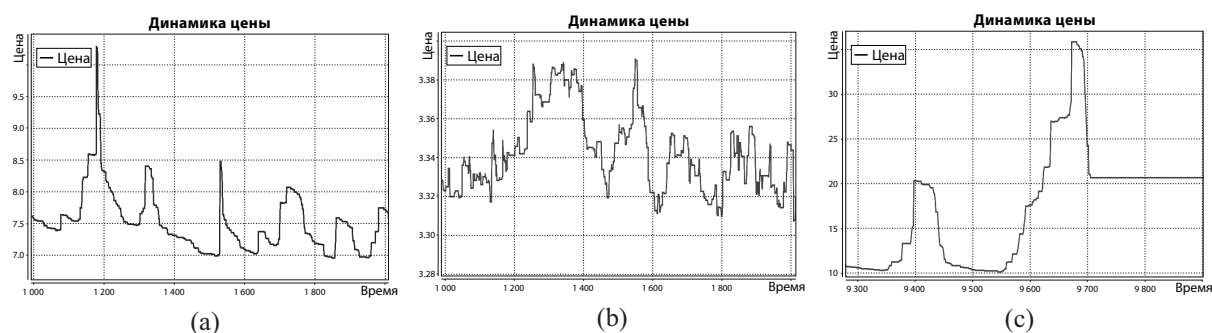


Рис. 2. Динамические режимы модели финансовой системы

В различных областях параметров модель демонстрирует широкий диапазон динамических режимов, включая экстремальные случаи – финансовые крахи (рисунок 2с). Кроме того, полученная динамика во многом схожа с реальной, в том числе в виду наблюдения типичных [Cont, 2001] для финансовых временных рядов статистических характеристик (подробнее в [Steryakov, 2012; Steryakov, 2012a]).

Преимущества использования агентного моделирования в данном случае проявились в возможности проводить исследования микродинамики, изучать процесс ценообразования, получать и исследовать распределения различных величин по агентам. Кроме того, была реализована возможность проведения пользовательских торгов, которые осуществляются с помощью добавления в систему агента, поведением которого может управлять оператор программы. Последнее позволило исследовать реакцию системы на внешнее воздействие (например, вброс и выкуп рискованных активов).

## ООМ для биологической системы

Вторая модель, построенная на основе предложенной архитектуры, описывает взаимодействие биологических популяций [Стеряков, 2012b; Стеряков, 2013]. Данная модельная система, сходная с известной моделью «Хищник–Жертва», взята в качестве примера, иллюстрирующего применение разработанной архитектуры. Моделируется поведение двух типов агентов: хищников и жертв. Первым для продолжения популяции необходимо наличие вторых, которые в свою очередь размножаются, имея неограниченный ресурс питания. Все агенты обладают конечной продолжительностью жизни. В пространстве каждый агент представляется материальной точкой на плоскости. Моделирование происходит в ограниченной области. Для обеспечения непрерывной динамики при полном исчезновении агентов одного типа добавляется один объект данного типа – такое введение необходимо из-за дискретности числа объектов в отличие, например,

от классической модели, где оперируют концентрациями, значения которых могут быть сколь угодно малы.

В соответствии с предложенной автором архитектурой описываются три основные части: агенты, заявки и стековое взаимодействие. В данной модели вектор состояния агента принимает следующий вид:

$$\bar{v}^i(t) = \{x^i(t), y^i(t), a_1^i(t), a_2^i(t), a_3^i(t)\}.$$

Здесь  $x^i(t), y^i(t)$  — координаты  $i$ -го агента в пространстве, а составляющие вектора  $a_1^i(t), a_2^i(t), a_3^i(t)$  отвечают за его физиологическое состояние (питание, возраст и размножение).

Функция преобразования состояния агента изменяет координаты агента, задавая уравнения движения как случайное блуждание вида:

$$\begin{cases} x^i(t + \Delta t) = \begin{cases} x^i(t) + \xi^i, & 0 < x^i(t) + \xi^i < 1, \\ x^i(t) - \xi^i, & \end{cases} \\ y^i(t + \Delta t) = \begin{cases} y^i(t) + \zeta^i, & 0 < y^i(t) + \zeta^i < 1, \\ y^i(t) - \zeta^i, & \end{cases} \end{cases}$$

где  $\xi^i, \zeta^i = N(0, \sigma^2)$  — нормально распределенные случайные величины. Кроме того, функция преобразования отвечает за независимое изменение физиологических переменных, задавая им фиксированное приращение:  $a_j^i(t + \Delta t) = a_j^i(t) + d_j$ ,  $j = \overline{1, 3}$ .

Поведенческая функция  $\bar{f}^i(\bar{v}^i)$ , которая по внутреннему состоянию агента формирует заявку  $\bar{b}^i$  для системы взаимодействия, определяется следующим образом. Пусть для каждой физиологической переменной определены соответствующие константы  $A_1^i, A_2^i, A_3^i$ , задающие ее пороговое значение, при превышении которого поведенческая функция на выходе выдает запрос на удовлетворение соответствующей физиологической потребности:

$$b_j^i = \begin{cases} 0, & \text{если } a_j^i(t) < A_j, \\ a_j^i(t), & \text{если } a_j^i(t) \geq A_j, \end{cases}$$

для каждого  $j = \overline{1, 3}$ . То есть на каждом шаге с помощью поведенческой функции будет формироваться вектор заявки вида  $\bar{b}^i = \{b_1^i, b_2^i, b_3^i\}$ , где  $b_1^i, b_2^i, b_3^i$  характеризуют запросы на соответствующие физиологические потребности  $i$ -го агента.

Стековое взаимодействие осуществляется в отдельных стеках, по которым распределяются заявки в зависимости от координат соответствующих агентов, что отражает локальность взаимодействия. В каждом стеке выполняется удовлетворение всех возможных заявок, после которого происходит изменение векторов состояния агентов. Кроме того, объектная ориентированность предполагает и *метадействия* для каждого объекта: прекращение его существования, то есть исключение из системы, а также появление новых объектов.

На рисунке 3 представлена динамика численности популяций, полученная в результате моделирования. Как видно из графиков, происходят колебания численности двух типов агентов с временным сдвигом относительно друг друга примерно на полфазы. Наблюдается значительное сходство поведения системы с реальными аналогами.

Применение АМ при моделировании биологических взаимодействий открывает дополнительные возможности для исследования. Если с помощью классической модели Вольтерра–Лотки можно лишь изучать изменение концентраций, то в соответствующей агентной модели появляется возможность отслеживать микродинамику отдельных объектов, возможно исследование различных распределений по агентам (например, распределения продолжительности жизни



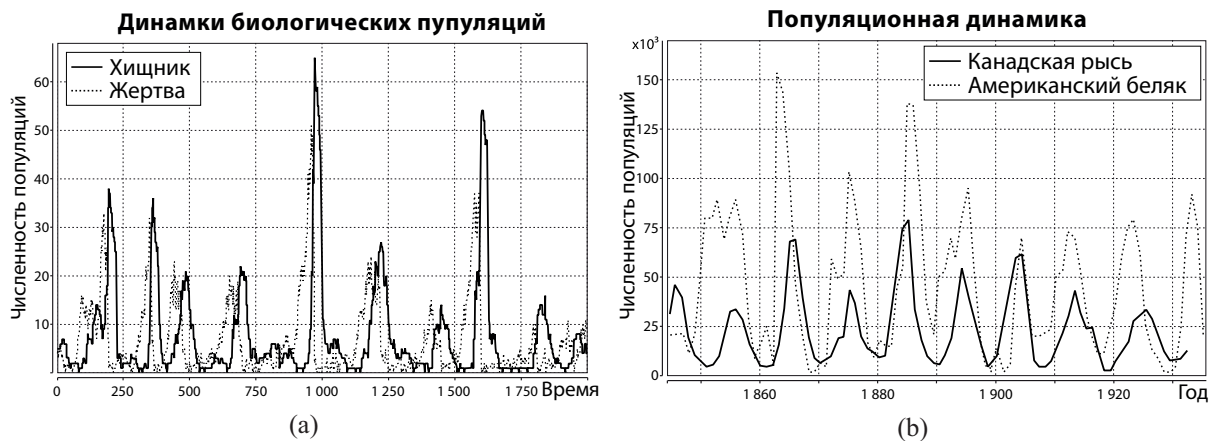


Рис. 3. Динамика численности двух конкурирующих популяций (а — результаты моделирования, б — статистика реальной системы)

жертв и хищников). Кроме того, отображая в каждый момент времени пространственные составляющие микросостояния системы, можно получить пространственное распределение популяций внутри территории. Пример временного среза для такого распределения представлен на рисунке 4.

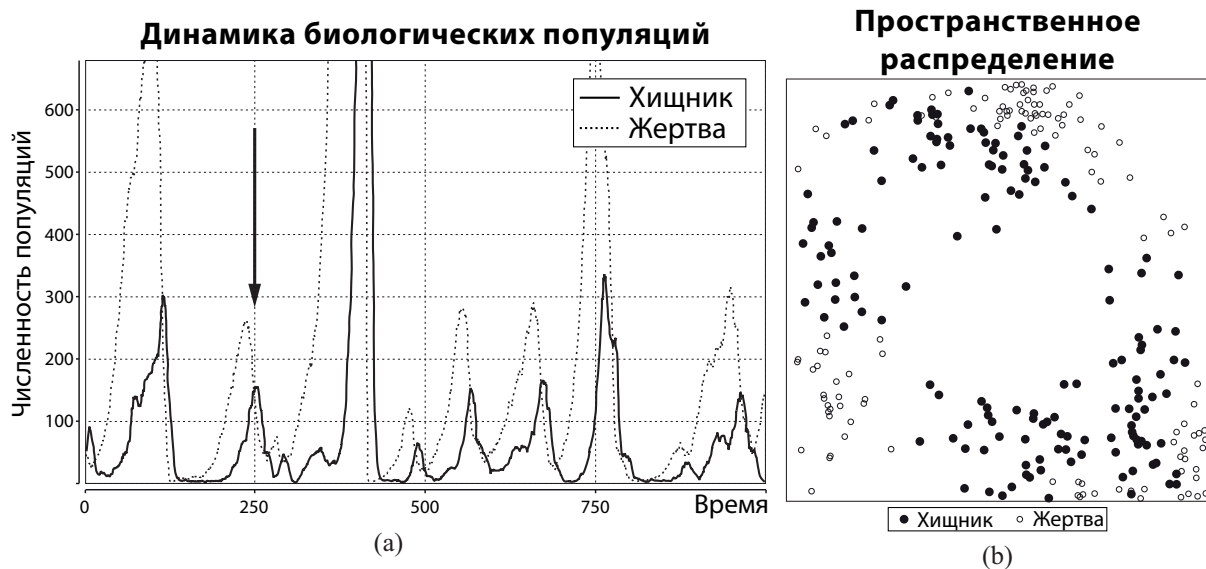


Рис. 4. Визуализация пространственно-временного распределения популяций

## Заключение

Приведенные примеры применения предложенной автором методики построения ООМ показывают, что объектная ориентированность модели позволяет существенно расширить ее описательные свойства: появляется возможность без глобальных изменений как модели, так и программной реализации вводить множество типов агентов или изменять поведение имеющихся, задавать внешние воздействия на исследуемую систему с помощью специальных агентов. Кроме того, описанная архитектура модели приводит к единообразию подхода к моделированию систем из различных областей.

Предложенный автором путь агентного моделирования отличается от большинства существующих инструментов создания моделей. Во-первых, с достаточной степенью точности описана методология подхода к моделированию, начиная с создания моделей и заканчивая реализацией программы для численных экспериментов. Таким образом, создание модели по заранее определенным правилам частично снимает проблему формализации. Во-вторых, метод предоставляет исследователю полную свободу действий в плане выбора средств визуализации, ввода любых дополнительных модулей в программу. Перечислим основные преимущества ООМ и предложенной архитектуры:

1. Использование языка программирования в качестве описательного инструмента позволяет моделировать любые объекты, любое поведение, не исключая сложные структуры в отличие от ограниченных средств различных сред моделирования.
2. Подход наследует от ООП повторное использование отдельных элементов программы. Например, модели агентов можно разрабатывать отдельно, объединяя потом в системы, в которых будет происходить взаимодействие.
3. Ясная связь в ООМ между моделью и ее компьютерной реализацией дает возможность передавать сформулированную объектно ориентированную модель специалисту в области программирования без дополнительных уточнений по реализации.

Как уже показала практика, агентные модели находят применение в самых различных областях знания. Существуют примеры удачного создания агентных моделей, построенных на принципах функционирования систем из других, не смежных областей: одним из таких примеров является применение к задаче маршрутизации муравьиных алгоритмов [Dorigo et al., 1996; Blum, 2005]. Таким образом, универсализация подхода к АМ может способствовать дальнейшему синтезу знаний в этом направлении, что в свою очередь открывает перспективу создания новых эффективных численных методов для решения практических задач.

## Список литературы

- Борщёв А. В.* От системной динамики и традиционного ИМ к практическим агентным моделям: причины, технология, инструменты // URL: <http://www.gpss.ru/paper/borshevarc.pdf>. — 2004.
- Городецкий В. И.* Самоорганизация и многоагентные системы. II. Приложения и технология разработки // Известия РАН. Теория и системы управления. — 2012а. — № 3. — С. 55–74.
- Городецкий В. И.* Самоорганизация и многоагентные системы. I. Модели многоагентной самоорганизации // Известия РАН. Теория и системы управления. — 2012b. — № 2. — С. 92–120.
- Замятина Е. Б., Чудинов Г. В.* Разработка и использование программных средств для построения и исследования агентных имитационных моделей // Вестник Пермского университета. — 2010. — № 2. — С. 80–84.
- Карпов Ю. Г.* Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. — БХВ-Петербург СПб., 2005. — С. 400.
- Сидоренко В. Н., Красносельский А. В.* Имитационное моделирование в науке и бизнесе: подходы, инструменты, применение // Междисциплинарный научно-практический журнал ГУ ВШЭ «Бизнес-информатика». — 2009. — С. 52.
- Стеряков А. А.* Математическая модель агентного типа для фондового рынка с неоднородными участниками // 18-я международная конференция «Математика. Компьютер. Образование». Под ред. Г. Ю. Ризниченко и А. Б. Рубина. — 2011. — С. 284.

- Стеряков А. А.* Имитационное моделирование фондовых рынков // Вестник Самарского государственного аэрокосмического университета. — 2012а. — № 2. — С. 274–281.
- Стеряков А. А.* Объектно ориентированное моделирование пространственно-временной динамики взаимодействия биологических популяций в ограниченной области // Третья международная конференция «Математическая физика и ее приложения»: Материалы конф. / под ред. член-корр. РАН И. В. Воловича и д. ф.-м. н. проф. В. П. Радченко. — Самара: СамГТУ, 2012б. — С. 284.
- Стеряков А. А.* Динамика сложных биологических систем в моделях агентного типа на примере взаимодействия двух конкурирующих популяций // 20-я международная конференция серии «Математика. Компьютер. Образование». Биофизика сложных систем. Анализ и моделирование. Тезисы. г. Пущино 28 января — 2 февраля 2013. — Москва–Ижевск, 2013. — С. 87.
- Blum C.* Ant colony optimization: Introduction and recent trends // Physics of Life reviews. — 2005. — Vol. 2, no. 4. — P. 353–373.
- Brock W. A., Hommes C. H.* Heterogeneous beliefs and routes to chaos in a simple asset pricing model // Journal of Economic dynamics and Control. — 1998. — Vol. 22, no. 8-9. — P. 1235–1274.
- Castle C. J., Crooks A. T.* Principles and concepts of agent-based modelling for developing geospatial simulations // Working papers series. Paper 110. — 2006.
- Chiarella C., He X.-Z.* Asset price and wealth dynamics under heterogeneous expectations // Quantitative Finance. — 2001. — Vol. 1. — P. 509–526.
- Cont R.* Empirical properties of asset returns: stylized facts and statistical issues // Quantitative Finance. — 2001. — Vol. 1. — P. 223–236.
- Dorigo M., Maniezzo V., Colonna A.* Ant system: optimization by a colony of cooperating agents // Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on. — 1996. — Vol. 26, no. 1. — P. 29–41.
- Levy H., Levy M., Solomon S.* Microscopic simulation of financial markets: from investor behavior to market phenomena. — Academic Press, 2000.
- Levy M., Levy H., Solomon S.* A microscopic model of the stock market: cycles, booms, and crashes // Economics Letters. — 1994. — Vol. 45, no. 1. — P. 103–111.
- Macal C. M., North M. J.* Tutorial on agent-based modeling and simulation // Proceedings of the 37th conference on Winter simulation. — WSC '05. — Winter Simulation Conference, 2005. — P. 2–15.
- Nikolai C., Madey G.* Tools of the trade: A survey of various agent based modeling platforms // Journal of Artificial Societies and Social Simulation. — 2009. — Vol. 12, no. 2. — P. 2.
- Politopoulos I.* Review and analysis of agent-based models in biology // University of Liverpool. — 2007.
- Steryakov A. A.* Agent-Based Model of the Stock Market // Market Risk and Financial Markets Modeling. — Springer, 2012. — P. 229–245.