

УДК 519.876.5

М.А. Бурилина

МЕТОДОЛОГИЯ МОДЕЛИРОВАНИЯ ПРОСТРАНСТВЕННЫХ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ НА ПРИМЕРЕ C#

*Центральный экономико-математический институт РАН,
Москва, Россия*

Остается несомненным, что развитие языков программирования, внедрения различных цифровых платформ и навигаций для усовершенствования визуализируемых моделей, в частности агент-ориентированных, приведет к необходимости создания единого языка программирования для суперкомпьютеров. Данная работа может быть полезна при разработке модели с участием интеллектуальных агентов, имеющих GIS-привязку к карте. Рассматриваемая модель построена с помощью языка программирования C#, включает в себя построение децентрализованных агент-ориентированных социальных систем. В работе описывается необходимость процесса распараллеливания для сложных агент-ориентированных моделей.

Ключевые слова: агент-ориентированные модели, искусственное общество, поведение агентов, архитектура агентов, распараллеливание, математическое моделирование поведения агентов, интеллектуальные агенты.

Развитие компьютерных технологий и суперкомпьютеров- является одним из рычагов развития и роста интереса к агент-ориентированным моделям (АОМ). Одно из основных применений модели- прогнозирование взаимоотношений между агентами в будущем периоде. Когда речь идет о комплексных процессах, в которые вовлечены различные группы агентов, общества, и даже страны (миграционные, экологические, политические процессы), то остается несомненным факт, что модель имеет разные полюса и веса, что делает ее сложной для прогнозирования и апробации. Именно мультиагентные модели могут справиться с таким комплексом задач. Важным признаком таких моделей является взаимодействие агентов, осуществляющих определенную деятельность в общей среде. Общепринятое определение [1] формулируется как набор особенностей объекта, которых достаточно, чтобы именовать его агентом. [2]

В настоящее время агент-ориентированный подход стал новым мощным направлением в социальном моделировании, применяемым во многих общественных науках. В основе данного подхода лежит взаимодействием отдельных сущностей, называемых агентами. Агентом является некий объект, воспринимающий информацию и действующий в определенной среде, у которого в ответ на последовательность актов восприятия, выполняются различные действия.

В работе [3] предложено обобщенное определение агента. Агент – представляет собой набор качеств, характеризующий рассматриваемую единицу или индивида, находящегося внутри своей окружающей среды и являющаяся ее частью, воспринимающая сигналы из окружения и

воздействующая на него для выполнения собственной программы действий.

Объектом исследования является интеллектуальный агент, а предметом исследования - среда, в которой агент взаимодействует с другими. Данная среда обладает различными свойствами, которые дальше рассмотрим более подробно и перейдем к определению и свойствам рассматриваемого объекта.

Для решения ряда экономических и демографических проблем и процессов в качестве агента в агент-ориентированных моделях выбираются интеллектуальные агенты. Они могут обладать следующими характеристиками:

1. Автономность – если они действуют без чьего-либо вмешательства, они сами контролируют свои действия и внутреннее состояние;
2. Социальность - они взаимодействуют с другими агентами с помощью некоторого языка, алгоритма взаимодействия и т.д.;
3. Реактивность - они воспринимают свое окружение, состоящее из других агентов, среды и т.д. и отвечают в режиме реального времени.
4. Целенаправленность (проактивность) – они не просто реагируют на окружение, но способны на целенаправленное поведение и проявление инициативы.

Архитектура интеллектуальных агентов довольно многообразна, существует ряд разновидностей: логическая, BDI (belief—desire—intention), реактивная, гибридная, включающая несколько архитектур. [4] Одна из самых распространенных именно гибридная, так как мы имеем дело со сложными поведением агентов, взаимодействующих между собой.

Теория игр - одна из отраслей экономики, где возможен анализ поведения агентов, принимающих рациональные решения. Интеллектуальные агенты в большинстве моделей экономического роста запрограммированы так, что принимают решения согласно равновесию по Нэшу: ни один из агентов не может увеличить свои преимущества в одностороннем порядке меняя свое решение, то есть стратегия обеих игроков является наилучшей реакцией на действия своего оппонента. В работе [5] помимо окружающей среды на действия агентов и равновесие принятия решений влияет «внешний агент», представляющий собой некий внешний механизм, устанавливающий «правила, форму игры или, что эквивалентно, выбирающего один из возможных механизмов взаимодействия участников, определяет специфику и, до некоторой степени, внутреннее системное противоречие рассматриваемой задачи». [5]. В частном случае «внешний агент» - это физическое или, как правило,

юридическое лицо, обладающее неким аппаратом власти на принятие решений у агентов-участников.

Рассмотрим далее частный случай моделирования агентов с помощью программного обеспечения Visual C Sharp.NET. (C#-обозначение языка программирования Visual C Sharp.NET) и опишем внешнюю среду агента. Причиной появления новых разнообразных программных продуктов и усовершенствованных языков программирования связано с многочисленными сложными задачами, которые решают с помощью АОМ. Для технической реализации агентов-ориентированных моделей на суперкомпьютерах растет и число программ, которые учитывают специфику таких моделей. Они могут быть разработаны для разнообразных целей и задач и учитывать особенности конструкции, логики поведения, архитектуру агентов, языки программирования, привязку к GIS- картам и другие. Тем не менее, необходимость создания новых специализированных решений остается по-прежнему актуальной задачей, особенно в тех случаях, когда АОМ используется как инструмент построения необычной и уникальной виртуальной среды [6, 7].

GISCoordinator-это интеллектуальная система-помощник, основанная на GIS. Она предназначена для аварийного реагирования на чрезвычайные ситуации, помогает IC (Incident Commander) координировать команду подразделений на местности, особенно в городских поисково-спасательных операциях. Эта пространственная интеллектуальная система поддерживает IC с интеллектуальными алгоритмами для планирования действий и планирования задач, чтобы централизованно координировать команду в динамичной и пространственной среде. Кроме того, она использует пространственную базу данных для управления географической информацией и информацией на основе определения местоположения и ГИС-функции для поддержки развития этих пространственных интеллектуальных алгоритмов. Язык программирования C Sharp был использован для реализации ядра GISCoordinator в работе [8]. В этой среде у агента имеется IC, оснащённый компьютером, на котором запущен GISCoordinator. Таким образом возникает сложная задача, в которой одновременно существует несколько команд, каждая со своим собственным IC. Для того чтобы максимизировать общую цель, команды должны сотрудничать и координировать свои действия, это и является основной задачей IC. В результате, распределенные IC образуют организацию или общество и должны согласовывать свои решения друг с другом в децентрализованном порядке. Что касается этого сообщества, GISCoordinator IC должен обеспечивать две важные возможности:

1) общаться и взаимодействовать с другими агентами, обмениваться данными и координировать решения,

2) использовать децентрализованные алгоритмы для координации распределенных решений. Человек и GICoordinator являются командой f человек- это агент (Incident Commander) Агенты связаны между собой сетью, которая называется СС (Communication Channel).

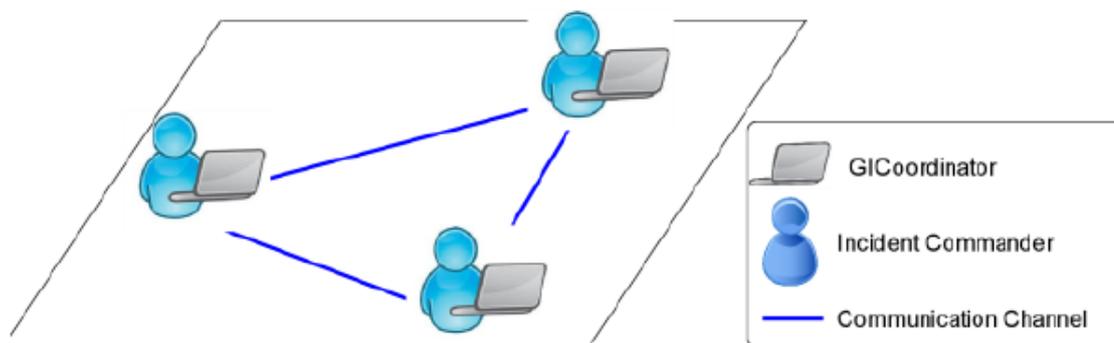


Рисунок 1. Взаимодействие между тремя агентами. Источник [8].

На Рисунке 1 иллюстрируется среда, в которой взаимодействуют три агента. Каждая микросхема оснащена GICoordinator, который запускается на планшетном компьютере. Распределенные в разных частях среды и имеющие территориальную привязку агенты-GICoordinator могут взаимодействовать друг с другом посредством передачи сообщений (данных).

Для расшифровки взаимодействий необходима программа, которая позволит реализовать и оценить две ключевые возможности GICoordinator прежде, чем могут быть разработаны реальные приложения. Следовательно, это необходимо, чтобы имитировать общество интеллектуальных агентов в заданной среде. Построение этой структуры имеет важное значение, поскольку она позволяет легко запускать несколько GICoordinators в нем, наблюдать за их поведением, проводить испытания и получать результаты GICoordinator. Агент-ориентированное моделирование (АОМ) и моделирование представляет собой относительно новый подход к моделированию систем, состоящих из автономных, взаимодействующих агентов. Симуляция организации GICoordinators имеет важное значение для разработки и оценки их эффективности работы организации. Симуляция поведения агентов на суперкомпьютере дает правила поведения агентов, в которых мы можем запустить ряд GICoordinators в виртуальном мире, изучить их взаимодействие и поведение при решении задач оптимизации распределения, и проверить эффективность распределенных алгоритмов координации. Это позволяет легко усовершенствовать архитектуру агентов, усложнять их взаимосвязи и перепроектировать в улучшенные алгоритмы. В интеллектуальных системах технологии, инфраструктура .NET может быть использована для

разработки агентов, таких как GICoordinator. Ниже мы рассмотрим какими требованиями должны отвечать агенты, при использовании языка программирования C SHARP на .NET. Мы имеем следующие требования (и допущения):

- связь между агентами осуществляется путем отправки и получения сообщений;
- симулятор и встроенные агенты работают на одном компьютере;
- каждый агент имеет доступ к центральной базе пространственных данных;
- человек-пользователь может выбрать любого агента, и взаимодействовать с ним;
- взаимодействие между агентом и человеческим-пользователем выполняется через интерфейс агента;
- мы можем инициировать и запускать любое количество агентов;
- есть как минимум три агента;
- все агенты имеют функции ГИС;
- для создания этого примера используется C SHARP .NET ;
- агент должен иметь возможность отправить определенное сообщение конкретному агенту;
- агент должен соответствующим образом реагировать на полученные сообщения.

Статья [8] представляет собой методологию для разработки агентного подхода для организации распределенных GICoordinators с использованием C #.NET в соответствии с требованиями, определенными выше. В описанном моделируемом обществе GICoordinator может обмениваться данными с другими агентами, отвечать на полученные сообщения, а также взаимодействовать с человеком-пользователем.

Агентно-ориентированное моделирование и моделирование (АОМ) представляет собой относительно новый подход к моделированию сложных систем, состоящих из взаимодействующих, автономных "агентов". Агенты имеют поведение, часто описываемое простыми правилами, а также взаимодействие с другими агентами, которые в свою очередь влияют на их поведение. Исследования АОМ могут быть отнесены к исследованиям сложных систем. Типичная агентно-ориентированная модель состоит из трех элементов:

1. набор агентов;
2. набор отношений и методов взаимодействия агента;
3. среда агентов.

Разработчик должен идентифицировать, смоделировать и запрограммировать эти элементы для создания сообщества на основе агентов. В целом, для моделирования агентных моделей доступны два типа

систем симуляции/моделирования: инструментальные средства и программное обеспечение. Инструментальные средства - это системы симуляции/моделирования, которые обеспечивают концептуальную основу для организации и проектирования агентных моделей. Они предоставляют соответствующие программные библиотеки, которые включают predefined процедуры и функции, специально предназначенные для агентного моделирования. Кроме того, агент-ориентированная парадигма позволяет интегрировать дополнительные функции, не предусмотренные в наборе инструментальных средств. Некоторые инструментальные средства включают Swarm, MASON, Repast, OBEUS и AnyLogic и другие программные обеспечения.

В дополнение к набору инструментов, программное обеспечение, такое как StarLogo, NetLogo и OBEUS доступно для разработки агентно-ориентированных моделей, и может упростить процесс внедрения. Например, используя программное обеспечение моделирования/симуляции часто можно избежать необходимости разработки агентно-ориентированной модели с помощью языка программирования низкого уровня (например, Java или C ++). В частности, программное обеспечение для АОМ является полезным для быстрой разработки моделей-прототипов. Тем не менее, разработчики с использованием определенного программного обеспечения ограничены рамками проекта, реализуемого программного обеспечения и конкретных задач. Например, некоторые программные продукты АОМ имеют ограниченную среду (например, только растровое), в которой моделируются или могут быть ограничены по размеру окружения агента. Кроме того, такая модель ограничена для функциональных возможностей, предоставляемых программным обеспечением (в отличие от наборов инструментальных средств ПРО, здесь не могут быть расширены или интегрированы дополнительные инструменты), особенно если инструментарий написан на своем собственном языке программирования (например, NetLogo). Большинство пакетов моделирования говорят, что они являются объектно-ориентированными или используют Java в качестве языка разработки.

Агент-ориентированная модель может быть запрограммирована полностью с нуля с использованием языка программирования низкого уровня (имеется в виду производительность), таких как Python, Java или C++. Однако развитие с нуля может быть неоправданно дорого, учитывая, что это требует развития многих услуг, уже представленных специализированными моделирующими агентами и набором инструментов. Большинство крупномасштабных агентных моделей используют специализированные инструменты, наборы инструментов или среды разработки из-за их удобства, простоты изучения, кроссплатформенной совместимости, а также необходимости для

возможности подключения сложных баз данных, графических пользовательских интерфейсов, а также ГИС. В частности, использование наборов инструментальных средств может уменьшить нагрузку на модельные устройства при программировании частей моделирования, которые не зависят от содержимого. Рассмотрим Таблицу №1, в которой произведен тест производительности языков программирования, подробнее можно ознакомиться здесь [9].

Таблица 1- Производительность языков программирования. Источник: [9].

Язык программирования	Java	C++	C++, -O2	PHP	Python 2.6	Python 3.1
Время исполнения, сек	14,3	8,5	2,6	62	139	208
Производительность, %	59,44	100	326,92	13,71	6,12	4,09

Однако, любой из языков программирования имеет свою цель, задачу, в которой применим лучше других. На Рисунке 2 представлена структура моделирования, основанная на требованиях, определенных в статье [7]. Одна из задач сосредоточена на моделировании организации в среде .NET. Для реализации этого были интегрированы поток и методы взаимодействия агентов, включенные в среду C SHARP .NET. Эти технологии позволяют легко дорабатывать архитектуру GICoordinator для достижения необходимых целей.

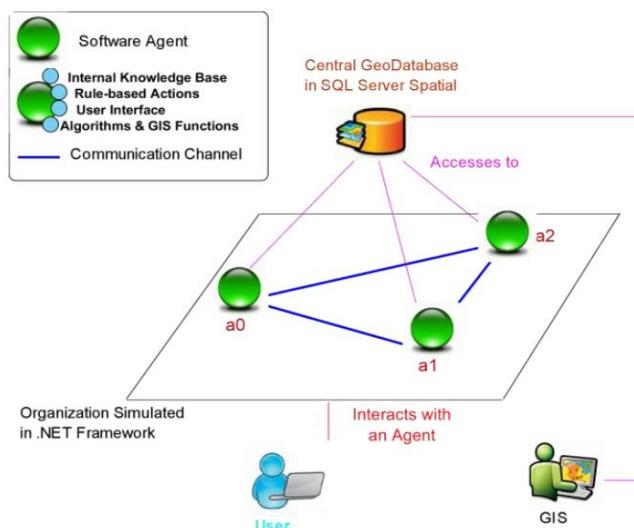


Рисунок 2. Структура моделирования. Источник: [8].

Обработка сообщений позволяет, программируя на C#, выполнять параллельную обработку, чтобы одновременно можно было выполнить несколько операций. C# поддерживает параллельное выполнение кода с помощью многопоточности. Поток - это независимый путь выполнения, который может запускаться одновременно с другими потоками.

Следовательно, можно писать приложения, которые выполняют несколько задач одновременно. Задачи, способные удерживать другие задачи, могут выполняться в отдельных потоках. Чтобы продемонстрировать это, в C SHARP была разработана базовая программа, демонстрирующая, как реализовать протокол контрактной сети (CNP) среди трех простых агентов в моделируемой организации. CNP - это протокол совместного использования задач в многоагентных системах, состоящий из набора узлов или программных агентов, которые образуют «контрактную сеть».

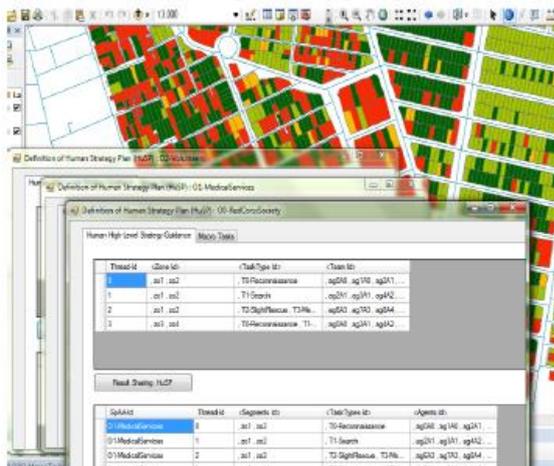


Рисунок 3. GIS- интерфейс взаимодействия агентов. Источник: [8].

На рисунке 3 представлены три окна агентов, где каждый GICoordinator имеет собственный пользовательский интерфейс, позволяющий агенту-пользователю выбирать и взаимодействовать между собой. Агенты отправляют сообщения, содержащие данные и информацию, которая требуется для координации их действий и решений. В качестве географической информационной системы, обеспечивающей географическую детализацию, используется программное обеспечение ArcGIS. Еще одной гео-системой для моделирования агентов служит AnyLogic, которая используется в модели [10] для прогнозирования транспортных заторов и взаимодействия частного и наземного государственного транспорта в г. Москве.

Когда узел получает сложную задачу (или по какой-либо причине не может решить ее текущую задачу), он разбивает проблему на подзадачи и объявляет подзадачу сети контрактов, которая выступает в качестве менеджера, представлено на Рисунке 4.

Рассмотрим частный случай взаимодействия экономического агента на примере закупок. В качестве агентов выступают агенты-заказчики и агенты-подрядчики. Тендерные заявки поступают от заказчиков к подрядчику (подрядчикам), выигравшим дело, которым вручается задание. Роль агента «a0» заключается в объявлении аукциона. Поскольку роль «a1» и «a2» заключается в заявке на этот аукцион (Рисунок 2), им не нужно

связываться друг с другом. Следовательно, мы не определили (установили) связь между этими двумя агентами «a1» и «a2». Таким образом, в случае отсутствия сговора между потенциальными подрядчиками, взаимодействие происходит только от агентов-заказчиков к агентам-подрядчикам. Почему в данной модели необходима гис- привязка? За частые транспортные затраты превышают в сумму со стоимостью заказа при подряде у дешевого агента-подрядчика превышают сумму итоговой сделки у более дорого агента-подрядчика. Поэтому стоимость доставки, выгрузки груза имеет большую значимость при принятии решения у агента-заказчика.

В дальнейшем планируется апробировать данную модель с гis-привязкой к городу Москве. В качестве агентов-заказчиков будут выступать государственные предприятия, а в качестве подрядчиков-агенты, зарегистрированные на государственном портале закупок «<http://zakupki.gov.ru>» и предлагающие свои услуги на тендерах.

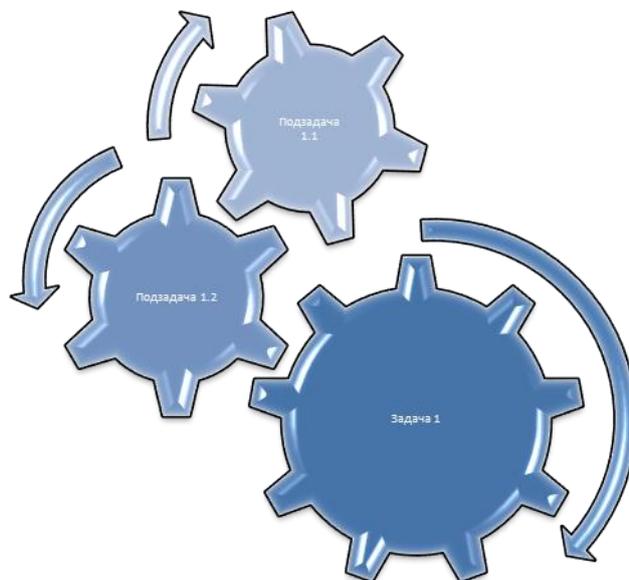


Рисунок 4. Разбиение процессов.

Чтобы построить работающую модель, в которой агенты действительно взаимодействуют и процессоров компьютера хватает для обработки их взаимоотношений необходимо применить эффективный инструмент для достижения требований, учитывая эти предположения. Существует довольно много литературы на базе агентного моделирования программного обеспечения, инструментальных средств и языков программирования, но, к сожалению, они не полностью решают проблемы, изложенные выше.

В работе рассмотрена архитектура интеллектуальных агентов. Описан процесс распараллеливания сложных взаимоотношений между агентами, выполненного по средствам языка программирования C#. Дана

описательная характеристика процессу, в котором агенты адресуются и взаимодействуют с друг другом с помощью передачи сообщений. Описана среда агентов, в которой потоки могут разделяться на операции агентов и их связь. Дано обобщенное условие для равновесия по Нэшу при взаимоотношении агентов.

Статья иллюстрирует использование языка программирования C SHARP для имитации модели: в искусственной среде социальный агент должен взаимодействовать с другими агентами и отвечать на полученные сообщения. Описанный подход заключается в интеграции процессов взаимодействия агентов, обеспечиваемых на платформе NET framework. *Исследование выполнено при поддержке Российского научного фонда (грант №14-18-01968).*

ЛИТЕРАТУРА:

1. «Intelligent Agents: Theory and Practice. M Wooldridge, NR Jennings» - The knowledge engineering review, Vol. 10:2, 1995, p. 115-152.
2. Бахтизин А.Р. Агентно-ориентированные модели экономики. М.: Экономика, 2008. 279 с.
3. Parsopoulos, KE «Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization». Natural Computing 1 (2-3): pp. 235–306. 2002. DOI:10.1023/A:1016568309421.
4. Бурилина М.А., Ахмадеев Б.А. Анализ многообразия архитектур и методов моделирования децентрализованных систем на основе агент-ориентированного подхода. Креативная экономика №7/2016 DOI: 10.18334/ce.10.7.35364
5. Вебер Ш., Давыдов Д., Савватеев А. Институты принятия решений // Вопросы экономики, № 6, Июнь 2017, С. 45-57.
6. В.Л. Макаров, А.Р.Бахтизин, Е.Д. Сушко, А.Ф. Агеева Искусственное общество и реальные демографические процессы, Экономика и математические методы, издательство Наука, том 53, №1, с.3-18.
7. А.Ф. Агеева Крупномасштабные агент-ориентированные модели и их техническая реализация на суперкомпьютерах. Моделирование, оптимизация и информационные технологии. Научный журнал №3(18) <http://moit.vivt.ru/> 2017 УДК 519.876.5.
8. Simulation of an Organization of Spatial Intelligent Agents in the Visual C#.NET Framework. Reza Nourjou and Michinori Hatayama International Journal of Computer Theory and Engineering, Vol. 6, No. 5, October 2014 DOI: 10.7763/IJCTE. 2014.V6.903 p.426-431

9. [Электронный ресурс] URL: <https://habrahabr.ru/post/66562/> (дата обращения: 20.09.2017).
10. Фаттахов М.Р. Агенто-ориентированная модель социально-экономического развития Москвы. Экономика и математические методы, 2013, том 49, N 2, с. 30–43.

M.A. Butilina

**METHODOLOGY OF SIMULATION OF SPATIAL INTELLIGENT AGENTS
AS EXAMPLE ON C #**

*Central Economics and Mathematics Institute of RAS,
Moscow, Russia*

It remains unquestionable that the development of programming languages, the introduction of various digital platforms and navigation to improve the visualized models, in particular agent-oriented, will lead to the need to create a single programming language for supercomputers. This work can be useful in developing a model involving intelligent agents that have a GIS-binding to the map. This model was built by using the C # programming language; witch includes the construction of decentralized agent-oriented social systems. The paper describes the need for a parallelization process for complex agent-based models. The article has been prepared with the support of the Russian Science Foundation, Grant No14-18-01968.

Keywords: agent-based models, artificial society, agent behavior, agent architecture, parallelization, mathematical modeling of agent behavior, intellectual agents.

REFERENCES:

1. «Intelligent Agents: Theory and Practice. M Wooldridge, NR Jennings» - The knowledge engineering review, Vol. 10:2, 1995, p. 115-152.
2. Bakhtizin A.R. Agent-oriented models of the economy. Moscow: Economics, 2008. 279 p.
3. Parsopoulos, KE «Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization». Natural Computing 1 (2-3): pp. 235–306. 2002. DOI:10.1023/A:1016568309421.
4. Butilina M.A., Akhmadeev B.A. Analysis of the diversity of architectures and methods for modeling decentralized systems based on the agent-based approach. Creative Economy No. 7/2016 DOI: 10.18334 / ce.10.7.35364
5. Weber Sh., Davidov D., Savvateev A. Institutions of decision making // Issues of Economics, No. 6, June 2017, pp. 45-57.

6. V.L. Makarov, A.R. Bakhtizin, E.D. Sushko, A.F. Ageeva Artificial society and real demographic processes, Economics and mathematical methods, publishing house Science, Volume 53, №1, p.3-18.
7. A.F. Ageeva Large-scale agent-oriented models and their technical implementation on supercomputers. Modeling, optimization and information technology. Scientific journal №3 (18) <http://moit.vivt.ru/> 2017 UDC 519.876.5.
8. Simulation of an Organization of Spatial Intelligent Agents in the Visual C#.NET Framework. Reza Nourjou and Michinori Hatayama International Journal of Computer Theory and Engineering, Vol. 6, No. 5, October 2014 DOI: 10.7763/IJCTE. 2014.V6.903 p.426-431
9. [Electronic resource] URL: <https://habrahabr.ru/post/66562/> (date of the address: 9/20/2017).
10. Fattakhov M.R. Agent-oriented model of social and economic development of Moscow. Economics and Mathematical Methods, 2013, Vol. 49, No. 2, p. 30-43.