

МНОГОАГЕНТНЫЕ СИСТЕМЫ: ОБЗОР СОВРЕМЕННЫХ ПОДХОДОВ К МОДЕЛИРОВАНИЮ И ПРОЕКТИРОВАНИЮ (Часть 2)

© 2024 М. А. Принев, Т. М. Леденева[✉], В. В. Гаршина

Воронежский государственный университет
Университетская пл., 1, 394018 Воронеж, Российской Федерации

Аннотация. Агент-ориентированные технологии позволяют выполнять сложные вычисления, решать многоуровневые задачи, осуществлять комплексное управление, имитировать реальные процессы, поэтому они имеют большое прикладное и практическое значение. Во второй части обзорного исследования рассматриваются различные подходы к моделированию многоагентных систем, современные направления их проектирования, приведены примеры инструментов разработки. Большое внимание уделено существующим приложениям многоагентных систем. Недостатком классического подхода к моделированию являются «жесткие» модели и заранее заданные протоколы коммуникации агентов, что не позволяет в полной мере реализовать такие свойства агентных систем, как самоорганизация, адаптация, способность к обучению и самообучению. Эволюционный подход базируется на организации вычислений на основе взаимодействий, при этом возникающие структуры требуют дополнительного анализа. Процесс разработки агентных приложений требует решения следующих основных задач: анализ предметной области и ее формализация; выбор модели многоагентной системы и формирование ее архитектуры; выбор модели агента, спецификация его свойств и поведения; формирование схем взаимодействия агентов, а также агентов и пользователей.

Ключевые слова: многоагентные системы, обобщенная концептуальная модель, классическое и эволюционное проектирование, паттерны проектирования, обучение, универсальные и прикладные платформы разработки многоагентных систем.

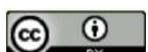
ВВЕДЕНИЕ

В настоящее время, обусловленное прорывным развитием цифровизации, искусственного интеллекта и распределенных систем, все большую актуальность получают исследования в области многоагентных систем (МАС), в рамках которых происходит интеграция достижений в вышеперечисленных сферах информационных технологий. С одной стороны, МАС — это открытая, активная, самоорганизующаяся система, парадигма которой позиционируется как подход к вычислениям на основе взаимодействий (англ. *computation as interactions*) [1, 2]. С другой стороны, МАС могут строиться по прин-

ципам распределенных вычислений, объединяя отдельные интеллектуальные системы, обладающие своими базами знаний и средствами рассуждений. Концепция интеграции посредством обмена асинхронными сообщениями, включающими диалоги и протоколы, оказалась востребована на практике и получила широкое распространение в других архитектурах и технологиях разработки сложных интеллектуальных систем.

При разработке МАС важнейшее значение имеет выбор модели и инструментов моделирования как агентов, так и всей системы. По сути, понятие *агента* базируется на понятии *объекта*, которое представляет собой «абстракцию множества экземпляров предметов реального мира, имеющих одни и те же свойства и правила поведения» [3]. В связи с этим многоагентная технология рассматривается

[✉] Леденева Татьяна Михайловна
e-mail: ledeneva-tm@yandex.ru



Контент доступен под лицензией Creative Commons Attribution 4.0 License.

The content is available under Creative Commons Attribution 4.0 License.

как новая парадигма программирования, в какой-то мере альтернативная объектно-ориентированному программированию.

В [4] приведены различные определения агентов, и каждый исследователь связывает понятие агента со своим конкретным набором свойств в зависимости от целей исследования. Важнейшим является понятие интеллектуального агента, который помимо выполнения определенных функций, связанных с решением задачи, должен иметь знания о себе и других агентах МАС, внешней среде, целях и способах их достижения, что интегрируется в базу знаний. Интеллектуальные агенты, имея представления о внешней среде и обладая возможностью рассуждать, способны предвидеть возможные реакции на свой действия, делать из этого выводы, полезные для дальнейших действий, и в результате прогнозировать свое состояние и состояние внешней среды.

Процесс моделирования МАС можно рассматривать как создание монолитной системы, обеспечивающей взаимодействие совокупности логических процессов, или же сделать акцент на программных компонентах, позволяющих создать сеть агентов для решения общих задач. При проектировании МАС должна быть решена задача синтеза целевой архитектуры, как это делается для распределенных систем [5], учитывая сложность и специфику задачи, поставленной для решения на основе агент-ориентированного подхода.

1. МОДЕЛИРОВАНИЕ И ПРОЕКТИРОВАНИЕ

Многоагентные системы — это инновационные информационные структуры, основанные на принципах коллективного взаимодействия автономных агентов с целью решения общих или индивидуальных задач.

МАС можно представить кортежем [6]:

$$MAC = (A, E, R, ORG)$$

где A — множество агентов, E — коммуникационная среда, R — множество взаимодействий между агентами, т. е. множество конфигураций, ORG — представление данной МАС как изображения.

В [7] предложена обобщенная концептуальная модель МАС, которая содержит сведения о структуре, поведении и семантике агентов, структуре и способах передачи сообщений между агентами, которые обеспечивают многопоточность и распределенность вычислений в системе. В общем случае модель МАС можно представить как совокупность субмоделей

$$M = (CNM, LPM, CLM),$$

где CNM — модель вычислительной сети (*Computing Network Model*), LPM — модель логических процессов (*Logical Processes Model*), CLM — модель концептуального уровня (*Conceptual Level Model*).

Модель вычислительной сети CNM является базовой и обеспечивает распределенность вычислений. Она определяет топологию сети в соответствии с требуемым объёмом необходимых вычислений и интенсивностью пересылки данных и представляется связным неориентированным графом, в котором вершинам соответствуют вычислительные узлы, а рёбрам — связи между ними с различными характеристиками (пропускная способность, латентность, таймаут соединения и др.). Данная субмодель определяется типом вычислительных систем и не зависит от предметной области или задач, которые должна выполнять МАС.

Модель логических процессов LPM обеспечивает связность субмоделей CNM и CLM . Она менее абстрактна, чем CNM , так как основными акторами системы являются логические процессы, которые непосредственно связаны с предметными задачами проектируемой МАС. Модель логических процессов можно представить следующим образом:

$$LPM = (t, S, L, M, T),$$

где t — множество моментов моделируемого времени; S — множество событий, при которых меняется состояние логического процесса; L — множество логических процессов; M — множество сообщений, передаваемых между логическими процессами; T — механизм передачи сообщений.

Множество логических процессов L представляет собой ориентированный граф, где

вершинами являются логические процессы, а дугами — связи между ними.

Сообщение из M представляется следующим образом:

$$m = (s, r, st, type),$$

где s — процесс-отправитель; r — процесс-получатель, st — передаваемое событие с временной меткой t , $type$ — тип сообщения.

Механизм передачи сообщений T представляет собой алгоритм передачи сообщений между логическими процессами в соответствующем временном режиме.

Модель логических процессов L проектируется в соответствии с принципами дискретно-событийного моделирования. Логические процессы изымают события из списков событий в соответствии с временной меткой t , а затем выполняется вычисление функции st , в результате логический процесс переходит из одного состояния в другое. В каждом логическом процессе различают три составляющие: статус, буфер входящих сообщений, буфер исходящих сообщений. Таким образом, модель логических процессов определяется несколькими параметрами, что делает ее менее абстрактной и повышает зависимость от конкретных задач МАС.

Модель концептуального уровня CLM является наименее абстрактной из субмоделей. Она зависит как от задач, поставленных для МАС, так и от предметной области в целом. На этом уровне происходит оперирование конкретными субъектами моделирования — агентами. Субмодель CLM можно представить следующим образом:

$$CLM = (AG, ME, Ont),$$

где AG — агентный граф; ME — среда моделирования; O — онтология предметного уровня.

Агентный граф AG строится в соответствии с задачами, которые должна решать проектируемая МАС, поэтому на его структуру не накладываются какие-либо ограничения. Различают абстрактный агентный граф (AAG), вершинами которого являются агенты МАС, а связи задаются статически в соответствии с задачами, которые должна решать система, и конкретный агентный граф (CAG), связи которого характеризуют текущие отно-

шения между агентами. AAG является декларацией количества, вида агентов и связей между ними. CAG имеет динамическую сущность и показывает состояние МАС в динамике происходящих процессов.

Среда моделирования ME определяется двумя множествами: множеством ресурсов (элементов произвольной природы, потенциально полезных для агентов) и множеством правил, определенных для среды моделирования (глобальных законов, описываемых на некотором языке). Для описания множества правил можно использовать правила на языке исчисления предикатов, функции распределения, системы алгебраических и дифференциальных уравнений, дискретно заданные функции, булевы функции, рекуррентные соотношения, утверждения на языке регулярных грамматик и т. д.

Онтология предметного уровня Ont представляется следующим образом:

$$Ont = (P, N, I),$$

где P — множество концептов; N — множество связей между концептами; I — способы интерпретации концептов посредством связей.

Онтология может использоваться для спецификации знаний, используемых интеллектуальными компонентами агентов, интеллектуального управления синхронизацией времени при распределённом моделировании, для управления распределением загрузки.

Моделирование МАС является нетривиальной задачей, однако, использование в процессе создания модели МАС принципа композиции наложения субмоделей, определенных на формальном уровне, позволяет систематизировать процессы проектирования. Например, при наложении субмоделей CNM и LPM вычислительные узлы связывают с логическими процессами посредством агрегации при выполнении следующих условий: на одном узле может выполняться несколько логических процессов; логические процессы могут быть связаны, если они находятся на одном узле или существует связь между узлами в CNM ; связи между логическими процессами должны быть направленными; граф

должен оставаться связанным. При наложении субмоделей *LPM* и *CLM* требуется связать агентный граф, окружающую среду и онтологию с множеством связанных логических процессов. Таким образом, принцип композиции наложения субмоделей заключается в проектировании свойств субмоделей друг на друга.

В общем случае можно выделить два основных подхода к моделированию и проектированию агентов и многоагентных систем — классический и эволюционный [7–10].

В рамках классической методологии каждая модель состоит из компонентов и взаимоотношений между ними, при этом модели разделяются на внешние и внутренние. Внешние модели относятся к системному уровню описания и, по сути, представляют собой абстрактные структуры агентов. Обычно выделяются два основных вида внешних моделей: *модель агентов* и *модель взаимодействий*, определяющая способы связи (коммуникации) между агентами.

Модели агентов порождают модели классов агентов и модели экземпляров агентов, которые связаны между собой отношениями наследования, агрегации и т.п. Внутренние модели описывают внутреннюю структуру агентов данного класса, являясь расширением объектно-ориентированных моделей (мнения и цели) и динамических моделей (планы). Типы агентов определяются множеством ролей. Поскольку одному типу может соответствовать несколько ролей, то целесообразно стремиться к нахождению разумного компромисса между простотой понимания функционального назначения агента и эффективностью его реализации.

Модель взаимодействия агентов состоит из множества протоколов, при этом протокол включает следующий набор атрибутов: назначение взаимодействия; инициатор взаимодействия; респондент, с которым осуществляется взаимодействие); входная информация для начала взаимодействия; выходная информация, предоставляемая респондентом в ходе взаимодействия.

Недостатком классической методологии являются «жесткие» модели и заранее заданные протоколы коммуникации.

Альтернативная методология моделирования МАС основана на идеях и технологиях эволюционного проектирования, которое предполагает использование теории самоорганизации для моделирования взаимодействия агентов, обучения и самообучения, адаптации агента к среде в рамках некоторой схемы эволюции. В [10] предложена общая методика эволюционного проектирования агентов и МАС, при этом генотип связывается с наследованием ресурса агентов-родителей. Эволюционное проектирование агентов предусматривает решение ряда принципиальных проблем, включая:

- анализ общих причин и движущих сил эволюции агентов в МАС;
- исследование средств и методов моделирования эволюционных процессов;
- определение причин и механизмов возникновения разнообразия типов агентов;
- исследование механизмов развития адаптационных свойств агентов.

Как отмечается, существует множество моделей эволюции, но не одна из них не свободна от недостатков, преодолеть которые возможно за счет гибридизации.

В [8] сформулирована идея создания агентств (семей) как единиц эволюционирующих многоагентных систем.

В рамках классического и эволюционного подходов к моделированию и проектированию агентов и МАС можно выделить несколько сформировавшихся и апробированных методов и технологий.

В [11] описан подход к формированию многоуровневой модели МАС на основе *миварной технологии*, под которой подразумевается способ создания баз данных и баз знаний с изменяемой структурой на основе дискретного информационного пространства без ограничений по объему и формам представления. Данная технология предусматривает эволюционное накопление и быструю адаптивную обработку информации. Мивар — это наименьший структурный элемент дискретного информационного пространства, представляет собой более общий формализм, чем семантические сети, онтологии и модель «сущность-связь». Результат

моделирования с использованием миваров может быть представлен в виде двудольного графа, связывающего два набора объектов: исходные объекты и результирующие объекты. Сети миваров могут быть реализованы на отдельных вычислительных системах или сервис-ориентированных архитектурах. МАС на основе миварной технологии имеет следующую схему реализации:

- 1) система получает запрос на решение какой-либо задачи;
- 2) агент-субординатор разбивает эту общую задачу на подзадачи и распределяет их между агентами-исполнителями;
- 3) после решения задач агентами-исполнителями агент-интегратор выбирает лучший вариант.

Каждый агент исполнительского уровня занимается решением «своей» задачи на определенной для него области знаний и по определенным для него алгоритмам. При использовании миварного подхода необходимо решить проблемы, связанные с распределением задач между агентами и объединением результатов. Предполагается, что выбор агентов для решения подзадач осуществляется на конкурсной основе. Отличительной особенностью миварного подхода является универсальность, что позволяет использовать его в различных областях (логический вывод на сети гиперправил; защита информации в автоматизированных системах управления; создание самоорганизующихся комплексов оперативной диагностики; взаимодействие групп мобильных роботов и т. д.).

Для моделирования процессов взаимодействия интеллектуальных агентов широко используется имитационное моделирование [12]. Общее управление поведением агентов осуществляется следующим способом: сенсоры сканируют состояния объектов внешней среды и изменяют модель окружения; решатель, используя модель окружения, активирует цели агента; планировщик для выбранной цели строит и выполняет план действий, управляя эффектором; эффектор, получая управляющие сигналы от планировщика, реализует определенные механизмы воздействия на внешнюю среду. Особое внимание

уделяется построению модели диалога и выбору языка общения агентов. В [13] исследована зависимость вероятности возникновения синергетического эффекта от степени сходства целей агентов.

В [14] для моделирования МАС и агентов используется теория паттернов, основу которой создал У. Гренандер. Паттерн — это образец или шаблон. В своей теории он использует паттерны четырех видов. К первому типу относятся объекты, называемые образующими. Они моделируют относительно простые объекты реального мира, обладающие связями. Образующие также могут соединяться с другими объектами (образующими). Из образующих, путем объединения их в группы конструируются регулярные конфигурации, являющиеся объектами второго типа. Объектами третьего типа являются изображения, получаемые следующим образом: сходные между собой регулярные конфигурации объединяются в множество, на котором определяются классы эквивалентности. Отношение эквивалентности, определяющее класс эквивалентности на множестве регулярных конфигураций, называется изображением. Четвертым типом объектов являются образы. В теории паттернов образом называется множество изображений, инвариантных относительно преобразований подобия. В данной статье теория паттернов используется для структурного и функционального описания агентов и МАС. Процесс преобразования входного потока данных в выходной является признаком образующего агента. Взаимодействия агента реализуются как связи образующих.

При разработке МАС для решения сложных задач не всегда имеющиеся технологии моделирования и проектирования содержат необходимые инструменты, что обуславливает разработку оригинальных подходов. В этой ситуации для проектирования уникальной архитектуры МАС целесообразно использовать шаблоны паттерны проектирования [15, 16], которые содержат в себе обобщенные принципы и рекомендации по проектированию систем и способам интеграции. Основная польза от использования шаблонов состоит в снижении сложности разработки за-

счёт готовых абстракций для решения целого класса проблем. Существуют различные классификации паттернов [15–17]. Остановимся на одной из них, которая включает следующие шаблоны [15]:

– *структурные шаблоны* позволяют ввести в разрабатываемую систему дополнительные элементы, расширяющие ее функциональные возможности, но не оказывающие сильного влияния на программную архитектуру (например, паттерны «ведущий–ведомый», «адаптируемый активный объект», «приоритет», «удаленный конфигуратор», «виртуальный кластер», «декомпозиция задачи», «обновление общего состояния»);

– *поведенческие шаблоны* реализуют различные аспекты работы агента: от выполнения им конкретной роли в МАС до совмещения нескольких ролей (например, паттерны «интерфейсный агент», «реактивный агент», «советник», «эмержентное общество»);

– *миграционные шаблоны* описывают различные способы организации миграции агентов (например, паттерны «маршрут», «клон», «частичное клонирование»);

– *коммуникационные шаблоны* предназначены для организации социального взаимодействие между агентами (например, паттерны «протокол», «брокер», «диспетчер событий», «мобильная сессия», «умное сообщение», «администратор»);

– *архитектурные (системные) шаблоны* задают различные виды архитектур агентов, МАС или элементов, закладывают жесткие программные ограничения функционирования разрабатываемой системы, с которыми предстоит считаться при внесении изменений (например, «виртуальная среда», BDI-архитектура агента, сети потребностей и возможностей);

– *защитные шаблоны* позволяют организовать безопасное функционирование системы, когда в качестве агентов могут выступать сторонние приложения и/или необходимо ввести дополнительный уровень безопасности, чтобы агенты не имели неавторизированного уровня доступа к ресурсам среды выполнения агентов (например, паттерны

«охрана агентства», «аутентификатор», «контроллер доступа»);

– *когнитивные шаблоны* предназначены для того, чтобы наделить агента или МАС различными способами распознавания собственных намерений или намерений других агентов в системе, описывая архитектуру их рассуждений (например, паттерны «намерение», «ощущение и вывод», «гибридный распознаватель»).

При разработке МАС необходимо обеспечить их оптимальность с учетом известных системных критериев.

В [18] рассматривается свойство *надежности* МАС, модель которой учитывает возможность объединения исполнительных механизмов в контейнеры. Агентные платформы являются исполнительными контейнерами для агентов, а агенты в свою очередь являются исполнительными контейнерами для задач. Разработанная модель резервированной МАС ориентирована на стратегию обеспечения отказоустойчивости, в том числе на основе введения избыточных множеств агентов и агентных платформ.

Для того, чтобы обеспечить высокую производительность МАС наряду с большим объемом обрабатываемой информации, необходимо использовать интегрированные распределенные системы. Такая система представляет собой совокупность независимых процессоров, в которой осуществляется распределение выполнения операций между входящими в нее программно-аппаратными модулями. Распределенная система предлагается пользователю в качестве единого инструмента [19, 20]. Существует несколько основных разновидностей интеграционных решений, таких, как передача файлов между узлами распределенной системы, общая база данных, удаленный вызов процедуры, обмен сообщениями. Задача разработчика получить оптимальный результат интеграции приложений, избежав при этом возникновения проблем, а именно, ненадежности сети передачи данных, гетерогенности приложений (несовместимость различных форматов данных, платформ и языков программирования),

низкой скорости передачи данных. Как показывает практика, наиболее оптимальным вариантом интеграции приложений является комбинированное решение с большим удельным весом использования обмена сообщениями в асинхронном режиме.

Обмен сообщениями в асинхронном режиме представляет собой технологию скоростного взаимодействия между приложениями в асинхронном режиме с гарантией доставки информации. Процесс осуществляется транспортировку по логистическому каналу, объединяющему программы и доступному для узлов связи, сообщений (пакетов данных, представляющих собой некоторую структуру данных, например, строку, байтовый массив или объект). Программа-поставщик отправляет сообщение и размещает его в канале, а программа-получатель адресно считывает из канала необходимую информацию. Алгоритм асинхронного обмена сообщений состоит из следующих этапов: создание, отправка, получение, обработка.

Принцип асинхронности позволяет добиться следующих преимуществ:

- поставщик не должен заботиться о сообщении после его размещения в канале, так как система обмена сообщениями гарантирует его доставку потребителю;

- возможность промежуточного хранения сообщения в канале сохраняет информацию на этапе отправки на компьютере поставщика, при доставке информация сохраняется на компьютере потребителя, причем процессы волнообразно повторяются, если для доставки сообщения требуется пройти несколько узлов связи;

- появляется возможность удаленного взаимодействия, платформенная и языковая интеграция, нагрузка на узлы связи регулируется, преодолевается рассогласованность во времени, что ведет к повышению надежности системы.

Таким образом, разработка интегрированных распределенных систем является оптимальным решением для обеспечения быстродействия и надежности работы МАС, имеющей модульную структуру. Использование распределенной системы с асинхронным об-

меном сообщениями позволит добиться гомогенизации приложений, обеспечить крос-платформенность и языковую интеграцию внутри системы. Удаленное взаимодействие модулей с возможностью регулирования нагрузки на узлы связи повышает вариативность использования аппаратных и технических решений для реализации функционала системы, так как интеграционные решения позволяют снизить требования к программному и аппаратному обеспечению при сохранении оптимальных показателей быстродействия, надежности и стрессоустойчивости системы в целом.

В [21] к одному из главных критерии эффективности МАС отнесена степень ее *интеллектуальности*, проявляющаяся в степени развитости признаков рационального поведения агентов. Предложена формула для определения степени интеллектуальности МАС. Сформулирована задача оптимизации состава МАС за счет увеличения числа агентов, обладающих большим количеством признаков рационального поведения.

Важнейшим свойством интеллектуальных агентов и МАС является способность к обучению и самообучению. Для МАС, ориентированных на функционирование в условиях неопределенности, большое значение имеет обучение с целью повышения *эффективности* управления. В процессе функционирования агенты наряду с локальными вознаграждениями за свои действия на каждом шаге получают некоторое отложенное итоговое вознаграждение после решения поставленной задачи. Возможны ситуации, при которых максимизация полученных локальных вознаграждений становится для агентов более привлекательным сценарием, в результате чего происходит зацикливание в локальных максимумах функции вознаграждения агентов, что влияет на время и скорость их обучения.

В [22] модель интеллектуальной иерархической МАС представляется кортежем, который помимо стандартных компонентов, содержит R — множество подкреплений, назначаемых агентам за переход из одного состояния в другое для заданного множества S состояний окружающей среды; а также буфер

опыта *B*, полученный другими агентами при решении поставленной задачи. Предложена методология проектирования таких систем, которая включает решение следующих основных задач для обучающихся агентов на основе парадигмы – :

- интеллектуальный анализ окружающей среды на основе данных, получаемых от систем наблюдения, с целью формирования опыта агентов, его формализации и передачи другим агентам;
- разработка механизма коллективного взаимодействия агентов при обучении с подкреплением на основе анализа окружающей среды;
- разработка метода обмена опытом между интеллектуальными агентами, работающими в группе;
- разработка критериев эффективности обучения и соответствующих моделей оценки.

Активность агентов подразумевает действия и макродействия. Переход агентов из одного состояния в другое формально описывается ориентированным графом переходов, в котором вершина — это состояние внешней среды, а дуги обозначают переходы, причем каждой дуге присваивается вес, отражающий, в том числе, усредненное вознаграждение за выполнение перехода по этой дуге. Каждой вершине присваивается метка, содержащая сведения об окрестных вершинах. После кластеризации вершин графа макродействие заключается в выполнении перехода между кластерами. На основе метода распространения меток предложен метод сегментации макродействий. Под *опытом* агентов подразумевается последовательность макродействий.

В [23] предложен алгоритм обучения с подкреплением для децентрализованной МАС, под которой подразумевается группа агентов, координирующих свой действия при решении поставленной задачи на равноправной основе. Алгоритм подразделяет агентов на группы в зависимости от области видимости и обучает случайному взаимодействию и обмену опытом. В централизованных системах имеются агенты-менеджеры и агенты-подчиненные. Здесь возникают задачи

обучения как для агентов подчиненных, так и для агентов-менеджеров различных уровней в иерархической МАС. В [24] предложен алгоритм обучения для сверхцентрализованной МАС, когда группа агентов рассматривается как единый агент с агрегированными функциями вознаграждения. При решении моделировании МАС с обучением можно выделить следующие задачи: разделение задачи на подзадачи с определением соответствующей функции вознаграждения (за задачу отвечает агент-менеджер); группирование агентов в зависимости от области их видимости и назначением им подзадач в зависимости от стратегии (назначение отдельной подзадачи каждому агенту или одна задача назначается всем); выбор режима работы (кооперация или конкуренция). Покрепление для агента-менеджера рассчитывается в зависимости от успешности агентов-подчиненных, при этом в зависимости от критериев эффективности оптимальные переходы сохраняются как положительный опыт.

2. ИНСТРУМЕНТЫ РАЗРАБОТКИ МНОГОАГЕНТНЫХ СИСТЕМ

Любая МАС состоит из программных агентов и платформы, которая поддерживает их взаимодействие. Существующие инструменты разработки МАС, как правило, включают средства разработки обоих компонент. Они делятся на два базовых класса [25]: инструменты, ориентированные на разработку МАС BDI-архитектуры, и инструменты, ориентированные на разработку агентов с явным описанием группового поведения. При использовании BDI-архитектуры поведенческая модель агента включает выбор цели, построение плана ее достижения и реализацию этого плана, причем все эти действия BDI-агенты должны выполнять самостоятельно на основе своей мотивационной модели. Групповое поведение возникает как результат взаимодействия агентов и обмена сообщениями между ними, при этом оно может оказаться непредсказуемым. BDI-архитектура ориентирована на описание модели индивидуального поведения агента, но не на описание группово-

вого поведения MAC в целом. Существует широкий класс приложений, где BDI-архитектура оказывается эффективной.

Другой класс архитектуры – это архитектура, ориентированная на явное описание группового поведения. В этом случае на основе содержательного анализа приложения формируется множество возможных сценариев группового поведения MAC, для каждого из которых определяются роли и протоколы, в соответствии с которыми эти роли должны взаимодействовать. Затем за каждой ролью закрепляется агент или множество агентов. Поведение агентов управляется событиями, источником которых могут быть изменение состояния агента, или сообщения других агентов, или сигналы, поступающие от внешней среды. Таким образом, в этой архитектуре групповое поведение MAC описывается явно, а функции самих агентов и их роли полностью определяются особенностями этого поведения.

В [1, 3, 7, 8, 26, 27] представлены распространенные методики разработки MAC, которые используются на этапах анализа и проектирования. В основе методологии GAIA [28] лежит понятие интеллектуального агента. На ее основе формируются модели агентов, модели ролей, модели взаимодействий, модели обслуживания, которые описываются в терминах, близких к объектно-ориентированному или компонентному программированию. Методология AgentUML [29] используется для разработки моделей различных компонент системы в форме диаграмм на языке AUML, который является расширением языка моделирования UML. Язык UML также используется в методологии PASSI, которая позволяет строить модели агентов и их сообществ. Методология INGENIAS также использует расширение UML и позволяет строить модель цели/задачи, модели агентов и систем агентов, модель окружения, модели взаимодействия. В методологии FAML [1] при разработке MAC выделяют уровень проектирования и уровень реализации, при этом каждому уровню соответствуют внутренняя и внешняя модели агента, что позволяет разделить моделирование агента и его внутреннее наполнение.

Методология ADELFE [30] используется для моделирования адаптивных MAC на основе модели сотрудничества интеллектуальных агентов. Необходимо заметить, что ни одна из перечисленных методологий не поддерживает полный цикл разработки MAC.

В [31] проведен обзор современных платформ разработки MAC. Универсальные платформы разработки MAC не привязаны к какому-либо классу прикладных задач, поддерживают стандартные архитектуры MAC, стандарты и спецификации разработки (парадигму BDI, протоколы FIPA). Многие из них поддерживают разработку распределенных и параллельных систем.

По условиям использования универсальные платформы можно разделить на платформы с открытым исходным кодом и коммерческие платформы.

К универсальным платформам с открытым исходным кодом относятся следующие:

- ActressMas (C#) — использует модель акторов, асинхронные операции .NET;
- Akka (Scala/Java) — использует модель акторов, в которой инкапсулируются состояния и поведение; реализует иерархическую структуру управления акторами;
- ASTRA (Java) — реализует агентно-ориентированный язык программирования AgentSpeak (L), поддерживает архитектуру BDI-агентов и стандарты FIPA;
- JADE (Java Agent Development Framework) (Java) — одна из старейших производственных систем для разработки MAC, которая имеет GUI-интерфейс и поддерживается библиотеками: JADEX для моделирования рациональных BDI-агентов, BDI4Jade для моделирования рассуждений, JENA для моделирования взаимодействия агентов с онтологиями предметных областей, TuCSoN и XKlaim для координации агентов и моделирования агентных иерархических сетей.

К универсальным платформам разработки MAC для коммерческого использования относятся следующие:

- AnyLogic (Java) — платформа, сочетающая профессиональные средства агентного моделирования, дискретно-событийное моделирование и системную динамику;

– ExtendSim (C++) — реализует методологию агентного моделирования для бизнес-целей;

– FlexSim (C++) — программное обеспечение для дискретно-событийного агентного моделирования и анализа, поддерживает 3D визуализацию моделей;

– JACK (Java) — среда разработки MAC на основе BDI-архитектуры;

– Simio с GUI-интерфейсом — среда разработки MAC, поддерживающая дискретно-событийное моделирование и системную динамику, предусмотрен анализ рисков в режиме реального времени;

– GoldSim (C++) — платформа для динамического моделирования и визуализации сложных систем в науке и бизнесе.

В отдельный класс выделяют *платформы, ориентированные на моделирование и симуляцию* MAC, снабженные дополнительными эффективными инструментами визуализации. Помимо перечисленных выше универсальных платформ разработки MAC для коммерческого использования, к другим известным платформам относятся следующие:

– Agents Assembly (AASM) — предметно-ориентированный язык, предназначенный для масштабируемого контейнерного моделирования;

– AgentPy (Python) — платформа для анализа данных, основанная на агентном моделировании;

– Agents.jl (Julia) — платформа, основанная на grid-средах для распределенного многоагентного моделирования в 2D и 3D;

– AgentScript (JavaScript) — платформа, основанная на семантике NetLogo с программированием на основе паттернов MVC (Model, View, Controller);

– DEVS Suite Simulator (Java) — платформа на основе симулятора Parallel DEVS для моделирования на основе клеточных автоматов;

– fjage (Java, Groovy) — платформа на базе JADE для моделирования дискретных событий в реальном времени;

– GAMA (Java/GAML) — платформа для пространственного моделирования в MAC;

– InsightMaker — среда агентного моделирования в браузере;

– JAS-Mine (Java) — платформа для дискретно-событийного моделирования с использованием агентных и микросимуляционных моделей;

– MASON (Java) — платформа дискретно-событийного агентного моделирования с 2D или 3D визуализацией.

– MASS — библиотека для реализации параллельных вычислений для многоагентных и пространственных симуляций (платформа поддерживает мобильность агентов);

– MESA — фреймворк для разработки агентов (предусмотрены классы агентов, определение пространства движения агентов на сетке); для организации действий агентов имеется планировщик; возможность запуска модели с различными настройками параметров обеспечивается инструментами сбора данных и модулями анализа данных).

Одной из технологий, позволяющих реализовать программные компоненты MAC, является HLA (High Level Architecture).

Помимо универсальных платформ для разработки MAC существуют платформы, ориентированные на решение различных классов задач и образующие прикладные системы многоагентного моделирования. Перечислим задачи, решение которых осуществляется такими MAC:

– *моделирование социальных процессов и систем* (разрабатываются различные сценарии, касающиеся человеческого взаимодействия и поведения целых социальных групп и сообществ (например, [32]), разработка программных обучающих систем (например, [33]);

– *моделирование транспортных систем* (моделирование и анализ трафика, управление светофорами, выбор маршрута, системы планирования движения транспорта, моделирование пассажиропотока, моделирование поведения толпы (например, [34,35])).

– *моделирование поведения робототехнических систем* (координация аппаратных и программных агентов, моделирование физических объектов (например, [36,37]));

– *моделирование окружающей среды и экосистем* (климатические модели, модели взаимодействия человека и природы, моделиро-

вание в экологии, биологии, эпидемиологии (например, [38,39]));

– *организационное моделирование* (планирование и составление расписаний, моделирование бизнес-процессов предприятий и организаций, взаимодействия «человек – агент» в корпоративных средах (например, [40]));

– *экономические исследования* (приложения в бизнесе, маркетинге и экономике (например, [20,29]));

– *медицинские приложения* (например, персонализированное здравоохранение и управление больницами (например, [41, 42]));

– *развлечения, фильмы и игровые среды* (например, индустрия видеоигр [43]);

– *распределенные вычисления* (распределение и согласование ресурсов в облачных вычислениях и коммуникация в слабосвязанных кластерах (например, [44–47])).

3. ПРИЛОЖЕНИЯ МАС

В этом разделе рассмотрим более подробно приложения многоагентного подхода к решению различных прикладных задач.

В настоящее время активные исследования на базе многоагентных систем проводятся в сфере индивидуальной и коллективной робототехники. Мобильные роботы используются для доставки грузов, патрулирования режимных объектов, исследования незнакомой окружающей среды. Известен широкий спектр робототехнических устройств от таких компаний как Яндекс, Google, Amazone, Starship Technologies и др. Во всех этих устройствах применяется интеллектуальная система управления, включающая ряд подсистем сбора и обработки информации. В робототехнических системах, как правило, используются программно-аппаратные иерархические многоуровневые МАС. Иерархическая структура управления строится на принципах декомпозиции глобальной цели на множество локальных, которые затем преобразуются в задачи и распределяются по группам и агентам, входящим в их состав, с возможностью контроля выполнения и перераспределения задач при необходимости [37]. Многоуровневые системы управления робототехнически-

ми комплексами предусматривают постановку цели для каждого уровня МАС, при этом часть управляющих решений принимается на каждом уровне, а часть делегируется другим уровням посредством циркуляции восходящих и нисходящих межуровневых потоков.

В [25] для управления системой малых космических аппаратов используется многоагентная концепция, предполагающая возможность одновременного контроля за большим количеством меняющихся источников информации в автоматическом режиме. Каждый агент — это процесс, обладающий определенной частью знаний об объекте управления и возможностью обмениваться этими знаниями с другими агентами. В данной системе предусмотрено несколько типов интеллектуальных агентов, каждый из которых предназначен для выполнения специальной функции:

– агент-координатор (менеджер проекта) обладает информацией о различных агентах, на основе чего обеспечивает согласованное взаимодействие агентов;

– агент доступа к базам данных или базам знаний в актуальной предметной области;

– поисковый агент предназначен для поиска необходимой информации в различных источниках;

– агент обработки информации, по сути, предназначен для решения задач с использование предопределенных методов;

– множество агентов для обслуживания пользователя (клиентский агент, интерфейсный агент, агент составления запросов к базе данных, агент исполнения запросов к базе данных);

– множество агентов, обеспечивающих работу МАС (агент системного администратора, агент безопасности сети и др.).

Одно из важных приложений МАС — обследование (или патрулирование) мобильными роботами инфраструктурных объектов.

Развитие транспортно-логистической сферы и роботизация складов требует активного применения технологий искусственного интеллекта для управления цепочками поставок и повышения производительности процесса обработки грузов. Актуальной является зада-

ча маршрутизации складских роботов, решение которой позволит им оптимально перемещаться внутри склада, избегая столкновений; доставлять конкретные товары в указанные места назначения; выбирать наиболее оптимальные маршруты. В [48, 49] рассматриваются алгоритмы для навигации роботов-доставщиков с использованием обучения с подкреплением [50, 51]. В качестве среды испытаний для демонстрации эффективности алгоритмов используется находящаяся в открытом доступе среда RWARE [52] со встроенным механизмом избегания коллизий.

Разработка многоагентных систем с коллективным поведением — одно из направлений в области искусственного интеллекта. К настоящему времени предложено и исследовано большое количество моделей MAC с различными архитектурами, в которых исследуется коллективное поведение [53, 54].

В [55] описана модель динамики онлайн-дискуссий в сети интернет с использованием MAC, представлены две многоагентные модели дискуссии, которые позволяют в полной мере учесть характеристики отдельных сообщений и наличие группы авторов с индивидуальными моделями поведения.

Для создания программной среды с целью исследования социальных систем предназначена система MANTA, которая основана на поведении муравьиной колонии [56]. Несколько типов агентов взаимодействуют друг с другом при помощи «стимулов-команд», позволяющих активировать «примитивы-действия» с целью решения поставленной задачи. В [57] описывается многоагентная система моделирования эволюции искусственных организмов на уровне генома Tiera. Главной особенностью модели Tiera является то, что программы изменяются путем случайных мутаций и перестановок, но остаются выполнимыми, хотя могут потерять свои функциональные характеристики и полезность.

На основе агент-ориентированного подхода разработана система US National Model [30], включающая данные обо всем населении США, которая позволяет предсказать последствия распространения различных заболеваний. Так, данная система позволила осущес-

твить имитацию последствий распространения вируса гриппа A(H1N1/09) в масштабах всей планеты. В России исследователи ЦЭМИ РАН и МГУ запустили на суперкомпьютере «Ломоносов» модель, имитирующую развитие социально-экономической системы РФ на протяжении последующих пятидесяти лет. Эта модель включает сто миллионов агентов, имитирующих социально-экономическую среду России [57].

В [26] предложена система MEMMAS (Macroeconomic Model of Multi-Agent Simulation) для моделирования реакций субъектов экономики РФ на возмущения параметров состояния внешнего рынка, в которой в качестве агентов выступают производственные отрасли экономики, совокупность домашних хозяйств и государство. Поведение агентов описывается алгоритмами расчета основных экономических показателей, а их взаимодействие осуществляется на основе согласования межотраслевого баланса. Еще одним примером данной направленности является система управления регионом с применением многоагентных технологий [57], которая включает MAC адресного взаимодействия населения и органов государственной власти в социальной сфере, MAC управления департаментом социального блока и интернет-портал для интеграции ресурсов и межведомственного взаимодействия.

В [27] отмечается растущий интерес к системам с программными агентами в качестве активных участников образующихся сообществ, для описания которых целесообразно использовать подходы на основе концепции MAC открытого типа. Известны исследования для систем смешанного состава, т. е. когда составляющими могут быть роботы, люди или команды людей. Такие системы называются *виртуальными организациями*. В [58] в рамках развития интеллектуальной робототехники рассматриваются подходы к организации взаимодействия людей и коботов (коллaborативных роботов) для выполнения единой глобальной целевой задачи в единой области рабочего пространства на основе многоагентной технологии.

Известны приложения МАС производственной направленности. В [40] представлена концептуальная модель архитектуры МАС для управления производством сетевого предприятия, позволяющая усилить интеграцию производственных и бизнес-процессов, повысить уровень их адаптивности, обеспечить надежность функционирования. МАС включает многоагентную платформу, реестр активных агентов продуктов и ресурсов, составляющих конкретное сетевое предприятие, а также сервис передачи сообщений между агентами. Предложено развитие роли агента платформы, на которого возлагаются функции не только управления программными агентами, но и администрирования бизнес-экосистемы.

В [59] исследуется поведение динамической транспортной сети, состоящей из дорог и перекрестков, при этом трафик интенсивности движения автомобилей зависит от колебания спроса (количество автомобилей, которые хотят проехать по данному участку дороги) и колебания емкости (наличие блокировки, например, ремонт дороги). Архитектура МАС состоит из следующих элементов: агенты автомобилей, агенты инфраструктуры и виртуальная среда. Каждое транспортное средство имеет своего агента, которому известна следующая информация об автомобиле: пункт назначения, текущее местоположение и скорость, возможные маршруты к пункту назначения. Агенты инфраструктуры располагаются на перекрестках и участках дороги, их основная задача — отслеживание информации о текущем состоянии дорожных элементов и сбор информации от агентов автомобилей о намерении ехать через определенную развязку или дорогу (если с течением времени эта информация не обновляется агентами автомобилей, то она перестает быть актуальной). Виртуальная среда — это графическое представление дорожной карты города. Агенты автомобилей исследуют среду и ищут маршруты, по которым автомобиль может доехать до пункта назначения. Агенты делают оценки качества маршрутов. Рассчитав эту информацию, агент автомобиля выбирает наиболее предпочтительный

маршрут. После этого агент автомобиля информирует агентов инфраструктуры о своем намерении, что позволяет другим агентам автомобилей использовать эту новую информацию для принятия новых решений. Для обмена информацией в системе используются легкие агенты, которые являются аналогами искусственных муравьев. Применяются два вида легких агентов: муравьи-намерения и муравьи-разведчики. В разработанной МАС каждый агент, принимая решение, изменяет внешнюю среду, а затем учитывает эти изменения для модификации своего решения с учетом решений, которые приняли другие агенты.

В [60] представлена концепция управления ресурсами с использованием многоагентных технологий.

В [61] описывается архитектура когнитивной многоагентной организационно-технической системы управления сложным техническим объектом. Эффективность таких систем зависит от распределения ограниченных ресурсов между выполняемыми задачами в жестких временных рамках с учетом динамически изменяющейся внешней среды. Система имеет иерархическое представление. Стратегический уровень отвечает за реализацию когнитивных функций (досрочное планирование, постановка целей и осуществление целенаправленного поведения, обучение как процесс формирования базы знаний). На тактическом уровне осуществляется постановка задач и управление ресурсами, а на техническом уровне задания выполняются. Для каждого уровня разработаны интеллектуальные агенты, причем, совокупность агентов более низкого уровня может входить в инфраструктуру некоторого агента более высокого уровня. Вводится понятие *когнитивного агента* с гибридной архитектурой, которая включает

- сенсорный компонент для обеспечения функции интерфейса;
- интеллектуальный компонент, который выполняет когнитивные функции;
- реактивный компонент, действующий в соответствии с правилами *если условие, то действие*;

– исполнительный компонент, который преобразует команды реактивного компонента в воздействия на окружающую среду.

Особый интерес представляет использование многоагентных технологий для организации вычислений, причем сформировалось такое научное направление, как вычислительный интеллект. В последние годы в области вычислительных технологий и коммуникаций возникает новая парадигма прикладных вычислений — «вычисление как взаимодействие», т. е. вычисление организуется среди вычислительных сущностей, объединённых в организацию, что приводит к новым формам понимания, разработки и управления вычислительными системами. Технология открытых МАС имеет необходимый потенциал для реализации данной парадигмы. Цель статьи [27] — разработать концепцию и принципы функционирования универсальной открытой многоагентной среды, в рамках которой смогут существовать и взаимодействовать различные МАС, построенные на основе разных многоагентных платформ.

В [44] МАС используется для организации функционирования распределенной вычислительной системы, при этом агенты выполняют следующие функции: конфигурирование распределенной вычислительной системы, распределение вычислительной нагрузки между компьютерами, оптимизация нагрузки в зависимости от вычислительной мощности компьютеров сети. Предполагается, что количество компьютеров в сети может изменяться. При их увеличении возникает ситуация, позволяющая уменьшить время решения задачи и повысить отказоустойчивость (живучесть) вычислительных процессов. Исследована эффективность использования распределенных вычислений на основе многоагентной системы.

В [45] на основе МАС предложен метод защиты распределенных вычислений в нестабильной вычислительной среде от угрозы получения ложного результата, который позволяет решить следующие проблемы:

– организовать распределенные вычисления на любом доступном количестве вычислительных узлов;

– динамически масштабировать вычислительные ресурсы при изменении используемой вычислительной среды;

– обеспечить более высокую степень информационной безопасности вычислительных процессов и результатов вычислений по сравнению с централизованной распределенной вычислительной системой.

В [46] представлена многоагентная система защиты информации в виде множества взаимодействующих агентов, размещенных на хостах компьютерной сети. В МАС имеются агенты нескольких типов, которые можно классифицировать следующим образом: агенты, связанные с обнаружением вторжения и реакцией системы защиты информации; агенты, связанные с восстановлением системы; агенты обучения, которые ответственны за адаптацию других агентов к реконфигурации компьютерной сети и новым типам атак; мета-агенты, управляющие процессами защиты информации и обеспечивающие координацию поведения и кооперацию агентов при решении задач защиты информации; агенты-демоны, предназначенные для анализа обрабатываемых сообщений. Предполагается, что компонент архитектуры многоагентной системы защиты информации, ассоциированный с некоторым хостом, может состоять из одного или нескольких экземпляров агентов каждого типа. Агенты-демоны играют роль сенсоров, информируя о поступающих сообщениях и выполняя их первичную обработку. Информация от демонов поступает к одному из агентов защиты, которые обрабатывают эти сообщения и, решая свою задачу, активизируют других агентов.

Известны разнообразные приложения самоорганизующихся МАС в области интеллектуальной поддержки функционирования компьютерных сетей, среди которых — управление использованием ресурсов сети, поддержка процессов адаптивной маршрутизации в сетях динамического состава и структуры, в частности, в сетях мобильных устройств, поддержка концепции повсеместных вычислений и самоконфигурация виртуальных (оверлейных) сетей и ряд других.

В [47] многоагентная технология используется в системах мониторинга – специализированных вычислительных системах, которые оверлейно развертываются поверх узлов системы, в интересах которой реализуется мониторинг. Примерами многоагентных систем мониторинга являются следующие [63]: система управления центром обработки данных (Data Center Infrastructure Management, DCIM), автоматизированная система управления дорожным движением (АСУДД), интегрированная система управления сетью связи (Integrated network management system, INMS) и другие.

В [64] приведены примеры многоагентных систем распознавания образов: SwarmNet — многоагентная система для распознавания объектов на изображениях, которая использует алгоритмы машинного обучения для определения объектов и затем распределяет эту информацию между агентами, которые обрабатывают данные; Multi-Agent Image Classifier (MAIC) — система для классификации изображений, которая разбивает изображение на несколько частей для обработки отдельными агентами; Agent-Based Image Recognition (ABIR) — многоагентная система для совместной обработки изображения несколькими агентами и определения его класса.

В [1, 65, 66] упоминаются системы дистанционного обучения, построенные на основе многоагентной технологии: MASPLANG (MultiAgent System PLAtaforma de Nueva Generación), MAGADI (on-line Multi-domain AGent based Adaptive eDucational environment developed on the conceptual results obtained in the IRIS system), I-Help (Intelligent Helpdesk), CBR-Tutor (Case-Based Reasoning Tutor). Даные системы являются адаптивными, а агенты выполняют роли персональных ассистентов, координаторов, инструкторов, планировщиков и др.

Многоагентные системы используются также и в медицинской диагностике. Примером такой разработки может служить MAC, предназначенная для построения системы интеллектуальной поддержки принятия решений на основе анализа и классификации рентгеновских снимков грудной клетки [42].

Таким образом, агент-ориентированные технологии являются перспективным направлением исследований и разработок, позволяя осуществлять сложные вычисления, решать многоэтапные задачи, реализовывать комплексное управление, имитировать реальные процессы, поэтому они имеют большое практическое значение и практическую ценность.

ЗАКЛЮЧЕНИЕ

Многоагентные системы являются востребованной технологией в современных условиях, которые характеризуются активным внедрением искусственного интеллекта практически во все сферы человеческой деятельности. Многоагентная технология — это технология распределенного искусственного интеллекта. Агент-ориентированный подход позволяет разрабатывать приложения для робототехники, логистики, управления предприятием, для интеллектуальной поддержки функционирования компьютерных сетей, для управления дорожным движением, моделирования и анализа реальных объектов и др. Моделирование и проектирование многоагентных систем, применение различных интеграционных решений являются важными аспектами для разработки MAC. Концепция агентов и MAC предполагает возможность, а, следовательно, способность соответствующей программы, самостоятельно реагировать на изменение среды функционирования. Агент-ориентированный подход позволяет отделить интеграционные решения от программирования агентов. Немаловажным преимуществом MAC является возможность эффективно решать проблемы взаимодействия множества компонент программы.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Городецкий В. И. Современное состояние и перспективы индустриальных применений многоагентных систем / В. И. Городецкий, О. Л. Бухвалов, П. О. Скобелев // Управление большими системами: сб. трудов. – 2017. – № 66. – С. 94–157.
2. Luck M. [et al.] Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing) // AgentLink. – 2005. <http://www.agentlink.org/roadmap>.
3. Янушко В. В. Анализ технологий, применяемых в многоагентных системах / В. В. Янушко, А.В. Далекин, С.Н. Еркин // Перспективные информационные технологии и интеллектуальные системы, 2006. – № 4. – С. 30–37.
4. Леденева Т. М. Многоагентные системы: модели агентов и постановки задач, обеспечивающие их характерные свойства (часть 1) / Т. М. Леденева, М. А. Принев // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2024. – № 3. – С. 50–73. <https://doi.org/10.17308/sait/1995-5499/2024/3/50-73>.
5. Топорков В. В. Модели распределенных вычислений / В. В. Топорков. – Москва : ФИЗМАТЛИТ, 2006. – 320 с.
6. Тараков В. Б. От многоагентных систем к интеллектуальным организациям / В. Б. Тараков. – Москва : Букинист, 2002. – 352 с.
7. Митраков А. А. Подходы к построению систем агентного моделирования. Режим доступа: <http://simulation.su/uploads/files/default/incomplete-mittrakov.pdf> (дата обращения 21.01.2024)/
8. Гладков Л. А. Разработка новых подходов к проектированию многоагентных систем / Л. А. Гладков, Н. В. Гладкова // Информатика, вычислительная техника и инженерное образование. – 2019. – № 2(35). – С. 16–23.
9. Гладков Л. А. Эволюционное проектирование как инструмент разработки многоагентных систем / Л. А. Гладков, Н. В. Гладкова // Известия ЮФУ. Технические науки. – 2019. – № 4(206). – С. 26–36.
10. Гладков Л. А. Эволюционное проектирование многоагентных систем / Л. А. Гладков, Н. В. Гладкова // Известия ЮФУ. Технические науки. – 2021. – №4(221). – С. 51–61.
11. Белоусова А. И. Подход к формированию многоуровневой модели мультиагентной системы с использованием миваров / А. И. Белоусова, О. О. Варламов, А. В. Острогух, М. Н. Краснянский // Перспективы науки. – 2011. – № 5(20). – С. 57–61.
12. Бурый А. С. Эволюция агентного моделирования. Часть 2. Имитационное моделирование / А. С. Бурый, В. А. Фролов, А. П. Кулян ица // Информационно-экономические аспекты стандартизации и технического регулирования. – 2023. – № 6(75). – С. 46–52.
13. Кириков И. А. Исследование эффекта самоорганизации поддержки принятия решения в компьютерных системах на примере многоагентных систем / И. А. Кириков, А. В. Колесников, С. В. Листопад // Вестник Российского государственного университета им. И. Канта. – 2010. – № 10. – С. 79–90.
14. Маторин С. И. Формализация многоагентных систем с помощью теории паттернов // С. И. Маторин, М. В. Михелев // Вестник национального технического университета «Харьковский политехнический институт». Серия: Информатика и моделирование. – 2008. – № 49. – С. 96–100.
15. Чернышев С. А. Классификация общих шаблонов проектирования мультиагентных систем / С. А. Чернышев // Программные продукты и системы. – 2022. – № 4. – С. 670–679. DOI:10.15827/0236-235X.140.670-679 .
16. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – Санкт-Петербург : Питер, 2001. – 368 с.
17. Juziuk J., Weyns D., Holvoet T. Design patterns for multi-agent systems: A systematic literature review. In: Agent-Oriented Software Engineering, 2014. – P. 79–99. DOI:10.1007/978-3-642-54432-3_5
18. Игумнов А. В. Оценка надежности резервированных многоагентных систем / А. В. Игумнов, С. Э. Сараджишвили // На-

- ука и образование: научное издание МГТУ им. Н. Э. Баумана. – 2014. – № 1. – С. 83–114.
19. Noorunnisa S., Jarvis D., Jarvis J., Rönnquist R. A Conceptual Model for Human-Agent Teams. In Agents and Multi-Agent Systems: Technologies and Applications 2021: Proceedings of 15th KES International Conference, KES-AMSTA 2021, June 2021. – Springer: Singapore, 2021. – P. 17–26.
20. Sato K., Sugawara T. Multi-Agent Task Allocation Based on Reciprocal Trust in Distributed Environments. In Proceedings of the Agents and Multi-Agent Systems: Technologies and Applications 2021 / Jezic, G., Chen-Burger, J., Kusek, M., Sperka, R., Howlett, R.J., Jain, L.C., Eds. – Springer: Singapore, 2021. – P. 477–488.
21. Мутовкина Н. Ю. Критерии эффективности формирования оптимального состава многоагентной системы / Н. Ю. Мутовкина, А. Ю. Клюшин, В. Н. Кузнецов // В сб.: Новое слово в науке: Сборник материалов III Международной научно-практической конференции. – 2017. – С. 139–141.
22. Дубенко Ю. В. Анализ иерархического обучения с подкреплением для реализации поведенческих стратегий интеллектуальных агентов / Ю. В. Дубенко, Е. Е. Дышкант, Д. А. Гура // Вестник компьютерных и информационных технологий. – 2020. – № 9. – С. 35–45.
23. Дубенко Ю. В. Алгоритм обучения с подкреплением для децентрализованных многоагентных систем, основанный на обмене опытом и обучении агентов случайному взаимодействию // Ю. В. Дубенко, Н. А. Рудешко // Вестник Воронежского государственного технического университета. – 2022. – № 18? Т. 14. – С. 30–36.
24. Дубенко Ю. В. Алгоритм коллективного взаимодействия интеллектуальных агентов в централизованных многоагентных системах / Ю. В. Дубенко // Вестник компьютерных и информационных технологий. – 2022. – № 10. – С. 30–42.
25. Гончаренко В. А. Методы построения кластерных космических систем на основе многоагентных технологий / В. А. Гончаренко, В. С. Забузов // Интеллектуальные технологии на транспорте. – 2022. – № 2(30). – С. 32–40.
26. Лукацкий А. М. Инструментальные средства и архитектура системы многоагентного моделирования / А. М. Лукацкий, К. В. Несытых // В сб.: Управление развитием крупномасштабных систем MLSD'2016 – Труды IX международной конференции. – 2016. – С. 203–211.
27. Харин И. Г. Открытая многоагентная среда: концепция и платформа / И. Г. Харин, А. В. Яновский // Информационные технологии и вычислительные системы. – 2014. – № 4. – С. 12–24.
28. Leitao P., Vrba P. Recent Developments and Future Trends of Industrial Agents // Proc. of «Holonic and Multi-Agent Systems for Manufacturing» (HoloMAS-2011), Vol. 6867 of the series Lecture Notes in Computer Science. Springer Verlag, 2011–2012. – P. 15–28.
29. Применение суперкомпьютерных технологий для моделирования социально-экономических систем // Экономика региона. – 2015. – № 2(42). – С. 301–312.
30. Epstein J. Modelling to contain pandemics / Joshua Epstein // Nature. – 2009, V. 460, P. 687. DOI:10.1038/460687a
31. Wrona Z., Buchwald W., Ganzha M., Paprzycki M., Leon F., Noor N., Pal C.-V. Overview of Software Agent Platforms Available in 2023. Information 2023. – № 14. – 348 p. <https://doi.org/10.3390/info14060348>
32. Макаров В. Л. Социальное моделирование — новый компьютерный прорыв (агент-ориентированные модели) / В. Л. Макаров, А. Р. Бахтизин. – Москва: Экономика, 2013. – 295 с.
33. Hossam S., Abdelkader H., Khedr A., Salem R. Developing Multiagent E-Learning System-Based Machine Learning and Feature Selection Techniques // Comput. Intell. Neurosci. 2022. 2941840.
34. Pniewski R., Sellin D., Stankevich K., Ganzha M., Paprzycki M. Applying Software Agents to Make City Traffic Management Smarter. In Proceedings of the Second International Conference on Information Management and Machine Intelligence; Goyal D., Gupta A. K., Piuri V., Gan-

- zha M., Paprzycki M. Eds.; Springer: Singapore, 2021. – P. 651–659.
35. Pniewski R., Stankevich K., Ganzha M., Paprzycki M. Modelling and Optimizing City Traffic Using an Agent Platform. In Proceedings of the 2nd International Conference on Artificial Intelligence: Advances and Applications; Mathur, G., Bundele, M., Lalwani, M., Paprzycki, M., Eds.; Springer Nature: Singapore, 2022. – P. 861–868.
36. Романов П. С. Модель и функционально-структурная схема интеллектуальной системы управления робототехнической системы / П. С. Романов // Вестник Коломенского института (филиала) Московского политехнического университета. Серия: Естественные и технические науки. – 2017. – № 10. – С. 118–130.
37. Степанов М. Ф. Исследование алгоритмов функционирования интеллектуальной системы управления разведывательным робототехническим комплексом / М. Ф. Степанов, А. М. Степанов, Д. Ю. Петров, О. М. Степанова // Экстремальная робототехника. – 2021. – Т. 1, № 1. – С. 136–142.
38. Bourceret A., Amblard L., Mathias J. D. Governance in social-ecological agent-based models: A review. Ecol. Soc. – 2021. – 26. DOI:10.5751/ES-12440-260238
39. Jager W. Using agent-based modelling to explore behavioural dynamics affecting our climate. Curr. Opin. Psychol., 2021. – № 42. – Р. 133–139.
40. Тельнов Ю. Ф., Казаков В. А., Данилов А. В., Брызгалов А. А. Разработка моделей производственных и бизнеспроцессов сетевых предприятий на основе многоагентных систем // Программные продукты и системы, 2023. – Т. 36, № 4. – С. 632–643.
41. Fazio M., Pluchino A., Inturri G., Le Pira M., Giuffrida N., Ignaccolo M. Exploring the impact of mobility restrictions on the COVID-19 spreading through an agent-based approach. J. Transp. Health, 2022, 101373. DOI:10.48550/arXiv.2102.08226
42. Akbari Z., Unland R. A Holonic Multi-Agent System for the Support of the Differential Diagnosis Process in Medicine; Universität Duisburg-Essen: Duisburg, Germany, 2021. DOI:10.17185/duepublico/73939
43. Booth J. Marathon Environments: Multi-Agent Continuous Control Benchmarks in a Modern Video Game Engine. arXiv 2019, arXiv:1902.09097. <https://doi.org/10.48550/arXiv.1902.09097>.
44. Ховансков С. А. Алгоритм организации безопасных распределенных вычислений на основе многоагентной системы / С. А. Ховансков, В. А. Литвиненко, В.С. Хованская // Известия ЮФУ. Технические науки. – 2016. – № 10(183). – С. 146–158.
45. Ховансков С. А. Методика защиты распределенных вычислений в многоагентной системе / С. А. Ховансков, В. А. Литвиненко, В. С. Хованская // Известия ЮФУ. Технические науки. – 2019. – № 4(206). – С. 68–80.
46. Городецкий В. И. Архитектура базовых агентов многоагентной системы защиты информации в компьютерных сетях / В. И. Городецкий, И. В. Котенко // Известия ТРТУ. – 2000. – № 2(16). – С. 38–51.
47. Рыкишин М. С. Подходы к разработке алгоритма взаимного информационного согласования интеллектуальных агентов в распределенной многоагентной системе мониторинга / М. С. Рыкишин // Моделирование, оптимизация и информационные технологии, 2023. – Т.11, № 3(42).
48. Денисова Д. А., Гаршина В. В. Навигация погрузчиков в логистике с использованием мультиагентного обучения с подкреплением / Информатика: проблемы, методы, технологии: Сборник материалов XXIV международной научно-практической конференции / под ред. Д. Н. Борисова; Воронеж, Воронеж. гос. ун-т, 14–15 февраля 2024 г. – Воронеж, ВГУ, 2024. – С. 656–664.
49. Денисова Д. А., Гаршина В. В. Алгоритмы мультиагентного обучения с подкреплением категории on-policy / Труды молодых учёных факультета компьютерных наук ВГУ. Выпуск 4 / под ред. Д. Н. Борисова; Воронежский государственный университет. – Воронеж, 2024. – С. 366–376.
50. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative. Режим доступа: <https://arxiv.org/abs/2006.07869> (дата обращения: 01.06.2024).

51. The Surprising Effectiveness of PPO in Cooperative Multi-Agent. Режим доступа: <https://arxiv.org/abs/2103.01955> (дата обращения: 01.06.2024).
52. RWARE: A multi-agent reinforcement learning environment. Github. Режим доступа: <https://github.com/semitable/robotic-warehouse> (дата обращения: 01.06.2024).
53. Городецкий В. И. Самоорганизация и многоагентные системы. I. Модели многоагентной самоорганизации / В. И. Городецкий // Известия РАН. Теория и системы управления. – 2012. – № 2. – С. 92–120.
54. Городецкий В. И. Самоорганизация и многоагентные системы. II. Приложения и технология разработки / В. И. Городецкий // Известия РАН. Теория и системы управления. – 2012. – № 3. – С. 102–123.
55. Бабкин Э. А. Моделирование динамики онлайн-дискуссий в сети интернет с использованием многоагентных систем / Э. А. Бабкин, Т. С. Бабкина, Б. И. Улитин // Бизнес-информатика. – 2018. – № 2(44). – С. 17–29.
56. Drogoul A., Corbara B. MANTA: New Experimental Results on the Emergence of (Artificial) Ant Societies. Режим доступа: https://www.researchgate.net/publication/2465419_MANTA_New_Experimental_Results_on_the_Emergence_of_Artificial_Ant_Societies (дата обращения 24.03.2024).
57. Умная обработка процессов. Режим доступа: <https://www.tadviser.ru/index.php> (дата обращения: 27.03.2024).
58. Галин Р. Р. Эффективное функционирование смешанной неоднородной команды в коллаборативной робототехнической системе / Р. Р. Галин, А. А. Широкий, Е. А. Магид, Р. В. Мещеряков, М. В. Мамченко // Информатика и автоматизация. – 2021. – Т. 20, № 6. – С. 1224–1253.
59. Claes R. A decentralized approach for anticipatory vehicle routing using delegate multi-agent systems / R. Claes, T. Holvoet, D. Weyns // IEEE Transactions on Intelligent Transportation Systems. – 2011. – V. 12, № 2. – P. 364–373.
60. Ризванов Д. А. Алгоритмы управления ресурсами в сложных системах с применением многоагентных технологий / Д. А. Ризванов // Вестник Уфимского авиационного технического университета. – 2013. – Т. 17, № 5(58). – С. 63–69.
61. Непряев А. А. Архитектура когнитивной многоагентной системы управления сложным техническим объектом / А. А. Непряев // Евразийское Научное Объединение. – 2020. – № 8-2(66). – С. 114–116.
62. Чернышев С. А. Проблемы мультиагентных систем и возможные пути их решения / С. А. Чернышев // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. – 2023. – № 3. – С. 231–241. <https://DOI:10.18137/RNU.V9187.23.03.P.231>
63. Юлейси Г. П., Холод И. И. Взаимодействие в многоагентных системах интеллектуального анализа данных / Г. П. Юлейси, И. И. Холод // Известия СПбГЭТУ «ЛЭТИ». – 2020. – № 3. – С. 18–23.
64. Исаев И. Д. Многоагентные системы, алгоритм распознавания образов интеллектуальными агентами / И. Д. Исаев // Вестник науки. – 2023. – Т. 4, № 11(68). – С. 651–659.
65. Пашкин М. П. Многоагентная интеллектуальная система дистанционного обучения / М. П. Пашкин // Труды СПИИРАН. – 2006. – Т. 1, № 3. – С. 126–137.
66. Julian V, Botti V. Multi-Agent Systems. Applied Sciences, 2019. – 9. – 1402. Режим доступа: <https://doi.org/10.3390/app9071402> (дата обращения 20.10.2023).

Принев Мечислав Александрович — аспирант 2-го курса кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета.

E-mail: metshislav@rambler.ru

ORCID iD: <https://orcid.org/0009-0004-9704-6896>

Леденева Татьяна Михайловна — д-р техн. наук, проф., кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета.
E-mail: ledeneva-tm@yandex.ru
ORCID iD: <https://orcid.org/0000-0002-3944-2266>

Гаршина Вероника Викторовна — к.т.н., доцент кафедры технологий обработки и защиты информации Воронежского государственного университета.
E-mail: garshina@cs.vsu.ru
ORCID iD: <https://orcid.org/0000-0001-6903-010X>

DOI: <https://doi.org/10.17308/sait/1995-5499/2024/4/167-190>

ISSN 1995-5499

Received 25.09.2024

Accepted 18.12.2024

MULTI-AGENT SYSTEMS: A REVIEW OF MODERN APPROACHES TO MODELING AND PROJECTING (Part 2)

© 2024 M. A. Prinev, T. M. Ledeneva✉, V. V. Garshina

*Voronezh State University
1, Universitetskaya Square, 394018 Voronezh, Russian Federation*

Annotation. Agent-based technologies make it possible to perform complex calculations, solve multi-level tasks, carry out complex management, simulate real processes, therefore they are of great applied and practical importance. The second part of the review study examines various approaches to modeling multi-agent systems, modern directions of their design, and provides examples of development tools. Much attention is paid to the existing applications of multi-agent systems. The disadvantage of the classical approach to modeling is the «rigid» models and pre-defined protocols of agent communication, which does not allow to fully realize such properties of agent systems as self-organization, adaptation, ability to learn and self-study. The evolutionary approach is based on the organization of computations based on interactions, while the emerging structures require additional analysis. The process of developing agent-based applications requires solving the following main tasks: domain analysis and its formalization; the choice of a multi-agent system model and the formation of its architecture; the choice of an agent model, the specification of its properties and behavior; the formation of interaction schemes for agents, as well as agents and users.

Keywords: multi-agent systems, generalized conceptual model, classical and evolutionary design, design patterns, learning, universal and applied platforms for developing multi-agent systems.

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. Gorodetsky V. I., Bukhvalov O. L. and Skobelev P. O. (2017) Current state and prospects for industrial applications of multi-agent systems. *Management of large systems*. No 66. P. 94–157.
2. Luck M. [et al.] (2005) Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing). AgentLink. <http://www.agentlink.org/roadmap/>
3. Yanushko V. V., Dalekin A. V. and Yerkin S. N. (2006) Analysis of technologies used in

✉ Леденева Татьяна Михайловна
e-mail: ledeneva-tm@yandex.ru

- multi-agent systems. *Promising Information Technologies and Intelligent Systems*. No 4. P. 30–37.
4. Ledeneva T. M. and Prinev M. A. (2024) Multi-agent systems: agent models and problem statements providing their characteristic properties (part 1). *Bulletin of the Voronezh State University. Series: System analysis and Information Technology*. No 3. P. 50–73. DOI 10.17308/sait/1995-5499/2024/3/50-73.
5. Toporkov V. V. (2006) Models of distributed computing. Moscow : FIZMATLIT. 320 p.
6. Tarasov V. B. (2002) From multi-agent systems to intelligent organizations. Moscow : Buki-nist, 2002. 352 p.
7. Mitrakov A. A. Approaches to the construction of agent-based modeling systems. <http://simulation.su/uploads/files/default/incomplete-mitrakov>
8. Gladkov L. A. and Gladkova N. V. (2019) Development of new approaches to the design of multi-agent systems. *Computer science, computing technology and engineering education*. No 2 (35). P. 16–23.
9. Gladkov L. A. and Gladkova N. V. (2019) Evolutionary design as a tool for developing multi-agent systems. *Bulletin of SFedU. Technical sciences*. No 4 (206). P. 26–36.
10. Gladkov L. A. and Gladkova N. V. (2021) Evolutionary design of multi-agent systems. *Bulletin of SFedU. Technical sciences*. No 4 (221). P. 51–61.
11. Belousova A. I. [et al.] (2011) An approach to the formation of a multi-level model of a multi-agent system using mivars. *Prospects of Science*. No 5 (20). P. 57–61.
12. Buriy A. S., Frolov V. A. and Kulyanitsa A. P. (2023) The evolution of agent-based modeling. Part 2 Simulation modeling. *Information and economic aspects of standardization and technical regulation*. No 6 (75). P. 46–52.
13. Kirikov I. A., Kolesnikov A. V. and Listopad S. V. (2010). Study of the effect of self-organization of decision support in computer systems on the example of multi-agent systems. *Bulletin of the I. Kant State University of Russia*. No 10. P. 79–90.
14. Matorin S. I. and Mikhelev S. I. (2008) Formalization of multi-agent systems using pattern theory. *Bulletin of the National Technical University "Kharkiv Polytechnic Institute". Series: Computer Science and Modeling*. No 49. P. 96–100.
15. Chernyshev S. A. (2022) Classification of common design patterns for multi-agent systems. *Software products and systems*. No 4. P. 670–679.
16. Gamma E. [et al.] (1994) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Publ. 416 p.
17. Juziuk J., Weyns D. and Holvoet T. (2014) Design patterns for multi-agent systems: A systematic literature review. In: *Agent-Oriented Software Engineering*. P. 79–99.
18. Igumnov A. V. and Sarajishvili S. E. (2014) Reliability assessment of redundant multi-agent systems. *Science and education: scientific publication of Bauman Moscow State Technical University*. No 1. P. 83–114.
19. Noorunnisa S. [et al.] (2021) Conceptual Model for Human-Agent Teams. In *Agents and Multi-Agent Systems: Technologies and Applications 2021: Proceedings of 15th KES International Conference, KES-AMSTA 2021*, Springer: Singapore. P. 17–26.
20. Sato K. [et al.] (2021) Multi-Agent Task Allocation Based on Reciprocal Trust in Distributed Environments. In *Proceedings of the Agents and Multi-Agent Systems: Technologies and Applications 2021*, Springer: Singapore. P. 477–488.
21. Mutovkina N. Yu., Klyushin A. Yu. and Kuznetsov V. N. (2017) Criteria for the Efficiency of Forming the Optimal Composition of a Multi-Agent System. In the collection: *New word in science: collection of materials. III Int. scientific-practical conf.* P. 139–141.
22. Dubenko Yu. V., Dyshkant E. E. and Gura D. A. (2020) Analysis of hierarchical reinforcement learning for the implementation of behavioral strategies of intelligent agents. *Bulletin of Computer and Information Technologies*. No 9. P. 35–45.
23. Dubenko Yu. V. and Rudeshko N. A. (2022) A reinforcement learning algorithm for decentralized multi-agent systems based on the exchange of experience and training of agents in random interaction. *Vestn. Voronezh. State Technical University Unita*. Vol. 14. No 18. P. 30–36.
24. Dubenko Yu. V. (2022) Algorithm of collective interaction of intelligent agents in central-

- ized multi-agent systems. *Bulletin of Computer and Information Technologies*. No 10. P. 30–42.
25. Goncharenko V. A. and Zabuzov V. S. (2022) Methods for constructing cluster space systems based on multi-agent technologies. *Intelligent technologies in transport*. No 2 (30). P. 32–40.
26. Lukatsky A. M. and Nesykh K. V. (2016) Instrumental means and architecture of the multi-agent modeling system. In the collection: *Management of the development of large-scale systems MLSD'2016 – Proc. ninth international. Conf.* P. 203–211.
27. Kharin I. G. and Yanovsky A. V. (2014) Open multi-agent environment: concept and platform. *Information technologies and computing systems*. No 4. P. 12–24.
28. Leitao P. and Vrba P. (2011–2012) Recent Developments and Future Trends of Industrial Agents. *Proc. of «Holonic and Multi-Agent Systems for Manufacturing» (HoloMAS-2011)*, Vol. 6867 of the series Lecture Notes in Computer Science. Springer Verlag. P. 15–28.
29. Application of supercomputer technologies for modeling socio-economic systems. (2015) *The economy of the region*. No 2(42). P. 301–312.
30. Application of supercomputer technologies for modeling socio-economic systems. (2015) *The economy of the region*. No 2(42). P. 301–312.
31. Wrona Z. [et al.] (2023) Overview of Software Agent Platforms Available in 2023. *Information*. 14. 348.
32. Makarov V. L. and Bakhtizin A. R. (2013) Social modeling – a new computer breakthrough (agent-oriented models) M. : Ekonomika. 295 p.
33. Hossam S. [et al.] (2022) Developing Multiagent E-Learning System-Based Machine Learning and Feature Selection Techniques. *Comput. Intell. Neurosci.* 2941840.
34. Pniewski R. [et al.] (2021) Applying Software Agents to Make City Traffic Management Smarter. In *Proceedings of the Second International Conference on Information Management and Machine Intelligence*. P. 651–659.
35. Pniewski R. [et al.] (2022) Modelling and Optimizing City Traffic Using an Agent Platform. In *Proceedings of the 2nd International Conference on Artificial Intelligence: Advances and Applications*. P. 861–868.
36. Romanov P. S. (2017) Model and functional-structural diagram of the intelligent control system of the robotic system. *Bulletin of the Kolomna Institute (branch) of the Moscow Polytechnic University. Series: Natural and technical sciences*. No 10. P. 118–130.
37. Stepanov M. F. [et al.] (2021) Study of functioning algorithms of an intelligent control system for a reconnaissance robotic complex. *Extreme Robotics*. Vol. 1, No 1. P. 136–142.
38. Bourceret A., Amblard L. and Mathias J. D. (2021) Governance in social-ecological agent-based models: a review. *Ecol. Soc.* 26. DOI: 10.5751/ES-12440-260238
39. Jager W. (2021) Using agent-based modelling to explore behavioural dynamics affecting our climate. *Curr. Opin. Psychol.* No 42. P. 13–139.
40. Telnov Yu. F. [et al.] (2023) Development of models of production and business processes of network enterprises based on multi-agent systems. *Software products and systems*. Vol. 36, No 4. P. 632–643.
41. Fazio M. [et al.] (2022) Exploring the impact of mobility restrictions on the COVID-19 spreading through an agent-based approach. *J. Transp. Health.* 25. 101373.
42. Akbari Z. and Unland R. (2021) A Holonic Multi-Agent System for the Support of the Differential Diagnosis Process in Medicine. Universität Duisburg-Essen: Duisburg, Germany.
43. Booth J. (2019) Marathon Environments: Multi-Agent Continuous Control Benchmarks in a Modern Video Game Engine. arXiv. arXiv:1902.09097. <https://doi.org/10.48550/arXiv.1902.09097>.
44. Khovanskov S. A., Litvinenko V. A. and Khovanskova V. S. (2016) Algorithm for organizing secure distributed computing based on a multi-agent. *Bulletin of the Southern Federal University. Technical sciences*. No 10 (183). P. 146–158.
45. Khovanskov S. A., Litvinenko V. A. and Khovanskova V. S. (2019) Methodology for protecting distributed computing in a multi-agent system. *Bulletin of the Southern Federal University. Technical sciences*. No 4 (206). P. 68–80.
46. Gorodetsky V. I. and Kotenko I. V. (2000) Architecture of basic agents of a multi-agent in-

- formation security system in computer networks *Bulletin of TRTU*. No 2 (16). P. 38–51.
47. Rykshin M. S. (2023) Approaches to the development of an algorithm for mutual information coordination of intelligent agents in a distributed multi-agent monitoring system. *Modeling, optimization and information technology*. Vol. 11, No 3 (42). P. 16–27.
48. Denisova D. A. and Garshina V. V. (2024) Loader navigation in logistics using multi-agent reinforcement learning. Informatics: problems, methods, technologies: collection of materials of the XXIV International Scientific and Practical Conference. edited by D. N. Borisov; Voronezh, Voronezh State University, February 14–15, 2024. Voronezh, VSU. P. 656–664.
49. Denisova D. A. and Garshina V. V. (2024) Algorithms of multi-agent learning with reinforcement of the on-policy category. Proceedings of young scientists of the Faculty of Computer Science at VSU. No 4. edited by D. N. Borisov; Voronezh State University. Voronezh. P. 366–376.
50. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative. <https://arxiv.org/abs/2006.07869>
51. The Surprising Effectiveness of PPO in Cooperative Multi-Agent. <https://arxiv.org/abs/2103.01955>
52. RWARE: A multi-agent reinforcement learning environment. Github. <https://github.com/semitable/robotic-warehouse>
53. Gorodetsky V. I. (2012) Self-organization and multi-agent systems. I. Models of multi-agent self-organization. *Proceedings of the Russian Academy of Sciences "Theory and Control Systems"*. No 2. P. 92–120.
54. Gorodetsky V. I. (2012) Self-organization and multi-agent systems. II. Applications and development technology. *Proceedings of the Russian Academy of Sciences "Theory and Control Systems"*. No 3. P. 102–123.
55. Babkin E. A. Babkina T. S. and Ulitin B. I. (2018) Modeling the dynamics of online discussions on the Internet using multi-agent systems. *Business Informatics*. No 2 (44). P. 17–29.
56. Drogoul A. and Corbara B. MANTA: New Experimental Results on the Emergence of (Artificial) Ant Societies. https://www.researchgate.net/publication/2465419_MANTA_New_Experimental_Results_on_the_Emergence_of_Artificial_Ant_Societies
57. Smart processing of processes. <https://www.tadviser.ru/index.php>
58. Galin R. R. [et al.] (2021) Effective functioning of a mixed heterogeneous team in a collaborative robotic system. *Computer Science and Automation*. Vol. 20, No 6. P. 1224–1253.
59. Claes R., Holvoet T. and Weyns D. (2011) A decentralized approach for anticipatory vehicle routing using delegate multiagent systems. *IEEE Transactions on Intelligent Transportation Systems*. Vol. 12, No 2. P. 364–373.
60. Rizvanov D. A. (2013) Algorithms for resource management in complex systems using multi-agent technologies. *Bulletin of Ufa Aviation Technical University*. Vol. 17, No 5 (58). P. 63–69.
61. Nepryaev A. A. (2020) Architecture of a cognitive multi-agent control system for a complex technical object. *Eurasian Scientific Association*. No 8-2 (66). P. 114–116.
62. Gorodetsky V. I. and Kotenko I. V. (2000) Architecture of basic agents of a multi-agent information security system in computer networks *Bulletin of TRTU*. No 2 (16). P. 38–51.
63. Yuleisi G. P. and Kholod I. I. (2020) Interaction in multi-agent systems of data mining. *Bulletin of ETU "LETI"*. No 3. P. 18–23.
64. Isaev I. D. (2023) Multi-agent systems, algorithm for pattern recognition by intelligent agents. *Bulletin of Science*. Vol. 4, No 11 (68). P. 651–659.
65. Pashkin M. P. (2006) Multi-agent intelligent distance learning system. *Proceedings of SPIIRAS*. Vol. 1, No 3. P. 126–137.
66. Julian V and Botti V. Multi-Agent Systems. Applied Sciences, 2019. – 9. – 1402. <https://doi.org/10.3390/app9071402>

М. А. Принев, Т. М. Леденева, В. В. Гаршина

Prinev Mechislav A. — 2nd year graduate student, Department of Computational Mathematics and Applied Information Technologies, Faculty of Applied Mathematics, Informatics and Mechanics, Voronezh State University.

E-mail: metshislav@rambler.ru

ORCID iD: <https://orcid.org/0009-0004-9704-6896>

Ledeneva Tatyana M. — DSc in Technical Sciences, Professor, Department of Computational Mathematics and Applied Information Technologies, Faculty of Applied Mathematics, Informatics and Mechanics, Voronezh State University.

E-mail: ledeneva-tm@yandex.ru

ORCID iD: <https://orcid.org/0000-0002-3944-226>

Garshina Veronika V. — CSc in Technical Sciences, Associate Professor, Department Data Processing and Protection Technologies, Voronezh State University.

E-mail: garshina@cs.vsu.ru

ORCID iD: <https://orcid.org/0000-0001-6903-010X>