

Математическое моделирование
Mathematical modeling

УДК 004.89

<https://doi.org/10.32362/2500-316X-2026-14-5-85-93>

EDN GBLESX



НАУЧНАЯ СТАТЬЯ

Об агентно-ориентированном подходе к организации и разработке распределенных интеллектуальных систем

Е.И. Зайцев[®], Е.В. Нурматова

МИРЭА – Российский технологический университет, Москва, 119454 Россия

[®] Автор для переписки, e-mail: zajcev@mirea.ru

• Поступила: 11.03.2026 • Доработана: 20.05.2026 • Принята к опубликованию: 13.07.2026

Резюме

Цели. Обоснование целесообразности и концептуализация агентно-ориентированного подхода к построению распределенных интеллектуальных систем, функционирующих в условиях неполноты исходных данных, высокой стохастичности и нестационарности внешней среды. Разработка формальных моделей агентов и методов их кооперации. Построение децентрализованной архитектуры, обеспечивающей адаптивность, масштабируемость и отказоустойчивость. Проектирование и реализация подсистемы распределенного хранения и обработки знаний.

Методы. Решение поставленных задач потребовало интеграции методов агентного моделирования, аппарата теории автоматов, марковских процессов принятия решений, нечеткой логики, искусственных нейронных сетей, алгоритмов многоагентного обучения с подкреплением, а также процедур экспертного оценивания, обеспечивающих верификацию и практическую адаптацию разработанных теоретических положений.

Результаты. Разработана архитектура многоагентной интеллектуальной системы, основанная на кооперации автономных программных агентов, которая обеспечивает децентрализацию управления, слабую связанность компонентов и масштабируемость вычислительной инфраструктуры. Предложена формальная математическая модель интеллектуального агента, интегрирующая символьные методы представления и обработки знаний и нейросетевые алгоритмы. Разработана подсистема управления базой знаний, обеспечивающая динамическую реконфигурацию информационных потоков, адаптивную оптимизацию схем хранения и обработки данных, интеллектуальное кеширование, семантическую согласованность и поддержку немоногологического вывода на основе распределенных знаний. Реализована подсистема динамической балансировки вычислительной нагрузки на основе коалиции системных агентов-диспетчеров.

Выводы. Агентно-ориентированная парадигма является эффективной методологической основой для построения распределенных интеллектуальных систем, предназначенных для решения задач в слабоформализуемых проблемных областях, характеризующихся неполнотой данных, неопределенностью и нестационарностью внешней среды. Децентрализация управления и распределенная организация базы знаний позволяют преодолеть фундаментальные ограничения монолитных интеллектуальных систем, а также обеспечить масштабируемость и высокую отказоустойчивость разрабатываемой системы. Реализованные в многоагентной интеллектуальной системе механизмы динамической оптимизации распределенной базы знаний и адаптивной балансировки вычислительной нагрузки позволяют не только решать интеллектуальные задачи, но и эффективно управлять собственными вычислительными и информационными ресурсами.

Ключевые слова: распределенная система, многоагентная система, программный агент, система представления и обработки знаний, многоагентное обучение с подкреплением

Для цитирования: Зайцев Е.И., Нурматова Е.В. Об агентно-ориентированном подходе к организации и разработке распределенных интеллектуальных систем. *Russian Technological Journal*. 2026;14(5):85–93. <https://doi.org/10.32362/2500-316X-2026-14-5-85-93>, <https://elibrary.ru/GBLESX>

Прозрачность финансовой деятельности: Авторы не имеют финансовой заинтересованности в представленных материалах или методах.

Авторы заявляют об отсутствии конфликта интересов.

RESEARCH ARTICLE

An agent-oriented approach to the organization and development of distributed intelligent systems

Evgeniy I. Zaytsev[@], Elena V. Nurmatova

MIREA – Russian Technological University, Moscow, 119454 Russia

[@] Corresponding author, e-mail: zajcev@mirea.ru

• Submitted: 11.03.2026 • Revised: 20.05.2026 • Accepted: 13.07.2026

Abstract

Objectives. The work set out to justify the expediency and provide a conceptualization of an agent-oriented approach to building distributed intelligent systems that operate under conditions of incomplete initial data, high stochasticity, and non-stationarity of the external environment. Formal models of agents and methods for their cooperation were developed as part of a decentralized architecture that ensures adaptability, scalability, and fault tolerance. A distributed knowledge storage and processing subsystem was designed and implemented.

Methods. In order to solve the problems, agent modeling methods were integrated with automata theory, Markov decision processes, fuzzy logic, artificial neural networks, multi-agent reinforcement learning algorithms, along with expert evaluation procedures to ensure the verification and practical adaptation of the developed theoretical provisions.

Results. The developed multi-agent intelligent system architecture based on the cooperation of autonomous software agents provides decentralized control, weak coupling of components, and scalability of the computing infrastructure. The proposed formal mathematical model of an intelligent agent integrates symbolic knowledge representation methods and neural network algorithms. A distributed knowledge base management subsystem was developed to provide dynamic reconfiguration of information flows, adaptive optimization of data storage and processing schemes, intelligent caching, semantic consistency, and support for non-monotonic distributed knowledge-based inference. A dynamic load balancing subsystem was implemented based on a coalition of system software agents.

Conclusions. The described agent-oriented paradigm is an effective methodological basis for building distributed intelligent systems aimed at solving problems in poorly formalized problem domains characterized by incomplete data, uncertainty, and non-stationary external environments. Decentralized management and distributed knowledge base organization allow overcoming the fundamental limitations of monolithic intelligent systems which use the complete and consistent knowledge base and the monotonous logical inference, as well as ensuring the scalability and high fault tolerance of the developed system. The mechanisms of dynamic optimization of a distributed knowledge base and adaptive balancing of computational load implemented in a multi-agent intelligent system can be used not only to solve intelligent problems, but also to effectively manage their own computational and information resources.

Keywords: distributed system, multi-agent system, software agent, knowledge representation and processing system, multi-agent reinforcement learning

For citation: Zaytsev E.I., Nurmatova E.V. An agent-oriented approach to the organization and development of distributed intelligent systems. *Russian Technological Journal*. 2026;14(5):85–93. <https://doi.org/10.32362/2500-316X-2026-14-5-85-93>, <https://elibrary.ru/GBLESX>

Financial disclosure: The authors have no financial or proprietary interest in any material or method mentioned.

The authors declare no conflicts of interest.

ВВЕДЕНИЕ

Разработка многоагентных интеллектуальных систем (МАИС) представляет собой перспективный путь расширения границ автоматизации на слабо формализуемые проблемные области, в которых для решения проблемы требуется одновременное выполнение нескольких интеллектуальных операций, таких как логический вывод на основе экспертных правил, модификация базы знаний, анализ и обработка больших объемов гетерогенных данных, адаптация к условиям неопределенности и изменчивости внешней среды [1, 2].

Отличительной особенностью агентно-ориентированной парадигмы является декомпозиция сложной проблемы на совокупность подзадач, решаемых автономными программными агентами, которые обладают специализированными, взаимодополняющими знаниями и навыками (когнитивными моделями), необходимыми для решения делегируемых им подзадач. Благодаря скоординированной кооперации программных агентов, объединяющих свои локальные результаты, знания и вычислительные ресурсы для достижения общей системной цели, возникает синергетический эффект, при котором совокупный результат деятельности агентов качественно превосходит сумму результатов их изолированного функционирования. Синергетическая связь агентов трансформирует распределенную обработку информации в коллективный интеллектуальный акт, позволяющий системе эффективно агрегировать данные, порождать новые знания и вырабатывать решения, которые принципиально не могут быть получены ни одним из агентов в отдельности.

Рациональность и адаптивность поведения прикладных агентов обеспечивается синтезом нескольких технологий искусственного интеллекта внутри единой когнитивной архитектуры, которая включает два ключевых компонента: искусственные нейронные сети и подсистему представления и обработки знаний.

Использование в МАИС нейронных сетей позволяет программным агентам адекватно реагировать на изменения окружающей среды, а также формировать компетенции, эмерджентно повышающие эффективность многоагентной системы в условиях неопределенности. Реализованные на базе нейронных сетей механизмы

многоагентного обучения с подкреплением (Multi-Agent Reinforcement Learning, MARL) [3–5] позволяют сообществу агентов в процессе взаимодействия со средой и друг с другом находить оптимальные или субоптимальные координационные стратегии. Многоагентное обучение с подкреплением является ключевым инструментом для адаптации к нестационарным условиям, где заранее заданные протоколы взаимодействия неэффективны [6–8].

Агентно-ориентированная модель представления и обработки знаний, реализованная в МАИС в виде семантически согласованной распределенной базы знаний (РБЗ) [1, 2], обеспечивает агентов актуальным контекстом и позволяет производить немонотонный логический вывод на основе распределенных по сообществу агентов знаний. В отличие от классических монолитных интеллектуальных систем, базирующихся на знаниях (knowledge-based systems, KBS), в которых единый интеллектуальный решатель оперирует априорно заданной полной и непротиворечивой базой знаний, в МАИС сложная задача решается сообществом взаимодействующих автономных агентов, каждый из которых обладает собственной моделью знаний (убеждений), и специализированными методами решения задач. Убеждения представляют собой пропозиции, в истинности которых агент уверен в определенной степени на основе имеющихся у него данных, но которые могут быть пересмотрены при поступлении новой информации. Модель убеждений превращает программного агента из пассивного исполнителя логических правил в активного исследователя среды.

В централизованных системах представления и обработки знаний процесс решения сложной задачи формализуется как поиск на графе редукции (графе типа И/ИЛИ). Для ускорения обработки графа редукции задачи для классических монолитных KBS разработаны параллельные алгоритмы поиска. Однако данные алгоритмы ориентированы на использование в вычислительных системах с единой разделяемой памятью (Shared Memory Processors), как правило, с симметричной мультипроцессорной обработкой (Symmetric MultiProcessing, SMP), что накладывает жесткие ограничения на масштабируемость системы.

В МАИС прикладные программные агенты, являющиеся источниками предметных знаний, могут размещаться как на одном, так и на разных

SMP-узлах распределенной вычислительной системы. Для распределенных вычислительных систем (мультикомпьютеров, кластеров, облачных сред) эффективной парадигмой является модель распределенного программирования на основе передачи сообщений (message-passing model). В МАИС взаимодействие и координация программных агентов осуществляются исключительно через асинхронный обмен структурированными сообщениями по заранее определенным протоколам. Это обеспечивает слабую связанность (loose coupling) компонентов, что является ключом к масштабируемости и отказоустойчивости системы.

МАИС способна не только решать интеллектуальные задачи, но и оптимально управлять собственными вычислительными и информационными ресурсами в условиях нестационарной внешней среды. В МАИС балансировка вычислительной нагрузки между вычислительными узлами не является статической, а реализуется как активный адаптивный процесс. За эту функцию отвечает коалиция системных программных агентов-диспетчеров. Системные агенты в режиме реального времени собирают метрики с каждого узла и динамически перераспределяют вычислительную нагрузку путем миграции микросервисов и прикладных агентов между узлами кластера.

Производительность МАИС в значительной степени зависит от эффективности доступа к РБЗ. Для минимизации времени отклика на запросы и адаптации к паттернам обращений в МАИС реализованы интеллектуальное кеширование фрагментов онтологий и результатов частых запросов, предварительная загрузка в кеш вероятно востребованных данных, динамическая репликация знаний. Существенный прирост производительности обусловлен также работой специализированных программных модулей, осуществляющих динамическую оптимизацию логической структуры РБЗ и ее адаптивное согласование с параметрами нестационарной среды.

АРХИТЕКТУРА МАИС

Архитектура МАИС базируется на принципе функционального разделения программных агентов. На верхнем (прикладном) уровне функционируют интеллектуальные агенты, ответственные за решение целевых задач предметной области. Нижний (системный) уровень объединяет сервисных агентов, которые осуществляют непосредственное взаимодействие с ядром операционной системы, драйверами оборудования и событийно-управляемыми микросервисами. Связность и кооперация между всеми компонентами системы, включая взаимодействие агентов и микросервисов, обеспечивается за счет иерархии специализированных программных

интерфейсов, поддерживающих как внутривещный (intra-process), так и распределенный (inter-process) обмен информацией.

Прикладные программные агенты верхнего уровня реализуют согласованный набор из четырех базовых функций работы со знаниями: логический анализ (Knowledge-Based ANalySis, KB_ANS), связанный с выполнением процедуры логического вывода для интерпретации событий и установления причинно-следственных связей; семантическая ассоциация (Knowledge-Based ASSociation, KB_ASS), позволяющая обнаружить и установить смысловые связи между сущностями в онтологии системы; сравнительный анализ (Knowledge-Based CoMParison, KB_CMP), используемый для сравнительной оценки событий и объектов по заданным критериям и метрикам; процедура спецификации (Knowledge-Based VALuation, KB_VAL) осуществляющая поиск в базе знаний и верификацию значений атрибутов объектов.

Процедура KB_VAL поддерживает механизмы как точных (четких), так и нечетких запросов, реализованных как расширение языка структурированных запросов SQL (Structured Query Language) с применением аппарата нечеткой логики. В отличие от четких SQL-запросов, представляющих собой жесткую спецификацию условий, которым данные должны соответствовать полностью, в нечетких SQL-запросах можно использовать лингвистические категории (например, «низкий», «средний», «высокий» и т.п.). Таким образом устраняется необходимость в субъективном и часто произвольном назначении жестких границ, что делает запросы более интуитивно понятными для человека, а также позволяет ранжировать результаты по релевантности, не отбрасывая пограничные значения, а учитывая их с соответствующей степенью принадлежности.

Каждый прикладной программный агент в МАИС функционирует как автономная интеллектуальная единица, способная к самостоятельному принятию решений и адаптации. Эта способность обеспечивается за счет реализации алгоритмов многоагентного обучения с подкреплением, формальной основой которого служит модель децентрализованного частично наблюдаемого марковского процесса принятия решений.

Используемая в архитектуре МАИС децентрализованная модель MARL позволяет использовать стандартные алгоритмы обучения с подкреплением на уровне каждого отдельного прикладного программного агента. В этой модели все ключевые функции – принятие решений, оценка действий и обновление стратегии (политики) – выполняются агентом локально, на основе его собственного, частичного наблюдения за средой (рис. 1).

Процесс обучения строится по канонической схеме: агент, находясь в определенном состоянии

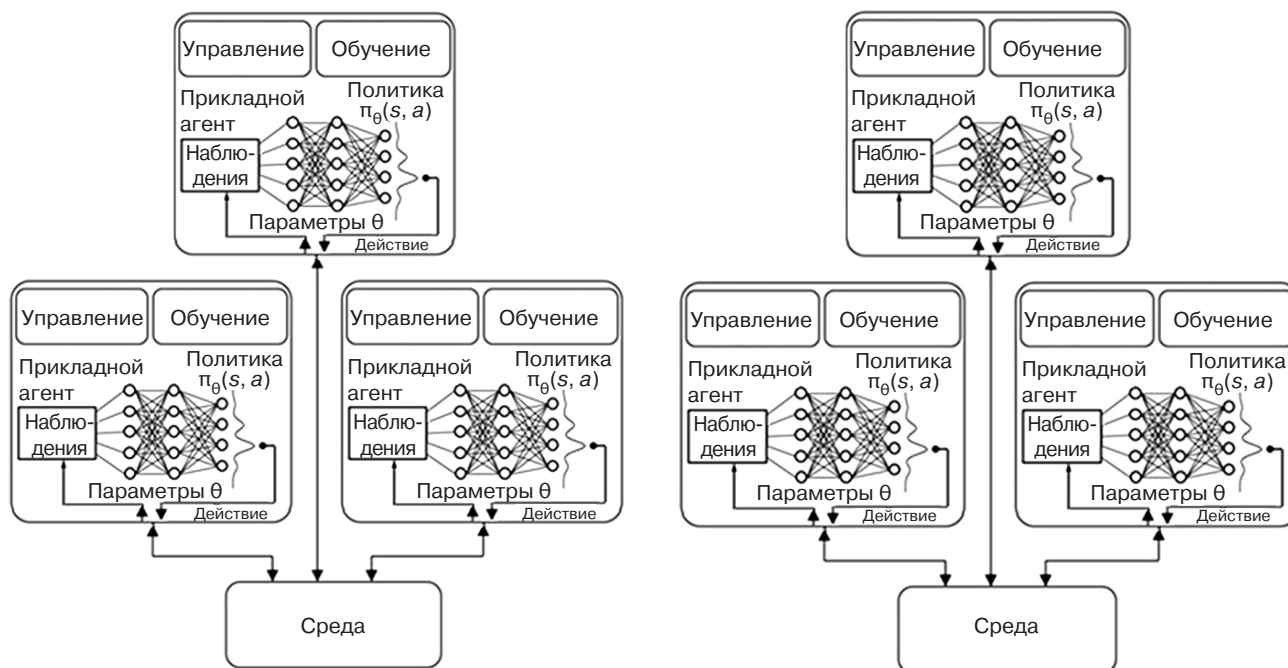


Рис. 1. Децентрализованная модель MARL.

Политика p_θ – стратегия действий агента, которая зависит от параметров нейронной сети θ ;
 s и a – состояния (states) и действия (actions) агентов соответственно

среды, выбирает и совершает действие. Это действие приводит к изменению состояния среды и порождает скалярный сигнал подкрепления (награду или штраф). Полученная награда служит оценкой полезности выполненного действия в данном контексте. В ходе итеративного обучения агент корректирует свою стратегию, стремясь максимизировать совокупное будущее вознаграждение, что в итоге формирует рациональное поведение, адаптированное к задачам системы.

В процессе совместного обучения в многоагентной среде существует вероятность достижения сообществом агентов равновесия по Нэшу. В данном состоянии ни один из агентов не может повысить свою индивидуальную полезность путем одностороннего отклонения от текущей стратегии, что может привести к стагнации и прекращению дальнейшего совершенствования стратегий.

Наиболее желательным исходом является ситуация, когда система достигает состояния, одновременно удовлетворяющего двум критериям: равновесию по Нэшу и оптимальности по Парето. В этой конфигурации стратегии агентов не только взаимно устойчивы, но и обеспечивают такое распределение выигрышей, при котором улучшение положения одного агента невозможно без ухудшения положения как минимум одного другого. Таким образом, агенты формируют коллективно оптимальные стратегии, находясь в устойчивом состоянии, где любое индивидуальное изменение стратегии невыгодно как для самого агента, так и для всей системы в целом.

Многоагентные интеллектуальные системы, базирующиеся на методах обучения с подкреплением, сталкиваются с необходимостью проектирования функций награды, с проблемами масштабирования и сложностью моделирования коммуникации на естественном языке. Внедрение в архитектуру МАИС больших языковых моделей (Large Language Model, LLM) приводит к качественному скачку в их функциональности. Благодаря наделению интеллектуальных программных агентов способностью понимать естественный язык, рассуждать и адаптироваться через контекстные инструкции, формируется принципиально новый класс адаптивных коллективных структур. Интеграция LLM в МАИС является одним из наиболее динамично развивающихся направлений современного искусственного интеллекта, трансформирующих подходы к координации, коммуникации и решению сложных задач [9–12].

В архитектуре МАИС коммуникация между прикладными программными агентами и сервисным слоем опосредована: запросы сначала поступают к системным агентам-диспетчерам, которые перенаправляют запросы доступным экземплярам соответствующих служб, обеспечивая тем самым балансировку нагрузки. В случае отказа одного из узлов кластера системные агенты исключают его из пула доступных ресурсов и перенаправляют весь трафик на работоспособные узлы. Механизм балансировки нагрузки, повышающий производительность и отказоустойчивость МАИС, иллюстрируется на рис. 2.

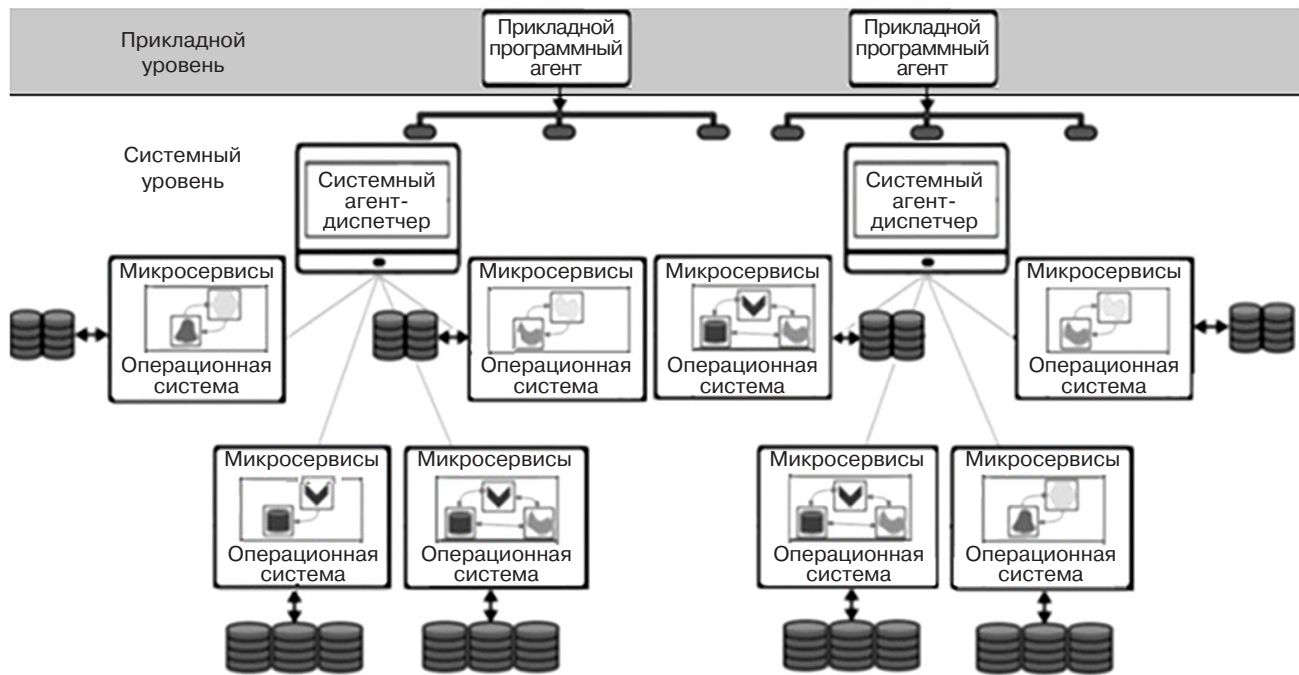


Рис. 2. Балансировка нагрузки в МАИС

Системные программные агенты, включенные в состав библиотеки операционной системы LibOS, которая напрямую взаимодействует с ядром операционной системы, реализуют также специализированную межпроцессную связь (InterProcess Communication, IPC) для передачи сообщений между вычислительными узлами. Специализированные программные модули LibOS менее требовательны к вычислительным ресурсам и обеспечивают более высокую производительность по сравнению с универсальными системными библиотеками.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ПРИКЛАДНОГО ПРОГРАММНОГО АГЕНТА

Математическая модель прикладного программного агента определяется следующим образом:

$$\text{Agent} = (X, S_{\text{ag}}, PL(D, s_0, G), PS, Y, A, M(X, S_{\text{env}}, S_{\text{KB}}(RL, Atr(V))), R_{\text{env}}, P_{\text{env}}),$$

где X – множество получаемых прикладным программным агентом входных сообщений; S_{ag} – множество состояний прикладного программного агента; $PL(D, s_0, G)$ – подсистема планирования в пространстве подзадач для проблемной области D с начальным состоянием агента s_0 и динамическим набором целей (желаний) агента G ; PS – продукционная система, определяющая переходы агента из одного состояния в другое; Y – множество выходных сообщений; A – набор действий агента, связанных с процессом обучения и поиском лучшей

стратегии; $M(X, S_{\text{env}}, S_{\text{KB}}(RL, Atr(V)))$ – множество функций (методов), определяющих реакцию агента (его действия) на входные сообщения X , наблюдения (состояние внешней среды) S_{env} , а также состояние S_{KB} , определяемое значениями выбранного из базы знаний набора атрибутов:

$$\text{intensional } RL_i = \{\dots[Atr_j, D(Atr_j)], \dots\};$$

$$\text{extensional } RL_i = \{B_1, \dots B_p\};$$

$$B_k = \{Atr_1(V_1), \dots, Atr_s(V_n)\}, k = 1, \dots, p,$$

где RL – набор отношений, хранящихся в базе знаний программного агента; $V = \{V_1, \dots, V_n\}$ – множество значений набора атрибутов Atr_j ; $R_{\text{env}}(S_{\text{env}}(t), a(t), S_{\text{env}}(t+1))$ – функция вознаграждения среды; $P_{\text{env}}(S_{\text{env}}(t+1)|S_{\text{env}}(t), a(t))$ – функция перехода состояний среды.

В случае, когда прикладной программный агент выступает в роли источника знаний и реализует четкие или нечеткие запросы к базе знаний, значения для атрибутов отношений выбираются из проблемной области D . Информация, характеризующая семантику проблемной области, содержится в интенциональной части базы знаний. В экстенциональной части РБЗ хранятся конкретные объекты, связи и состояния S_{KB} .

Информацию о функции переходов P_{env} и функции вознаграждений R_{env} прикладные программные агенты получают через состояния среды, действия и вознаграждения $r_i(t)$, т.е. через кортежи $(S_{\text{env}_i}(t), a_i(t), r_i(t))$, поступающие в момент времени t .

Функция переходов P_{env} представляет агенту информацию о внешней среде. Сформировав эту функцию, агент становится способным предсказывать новое состояние $S_{env}(t+1)$, в которое перейдет среда из состояния $S_{env}(t)$ после выбора агентом действия $a_i(t)$. Аппроксимация функций реализована в МАИС на основе нейронных сетей, которые моделируют как политику, так и функции ценности.

Если среда, в которой работает прикладной программный агент, известна и полностью предсказуема, то агент не нуждается в обучении, он будет действовать корректно в соответствии со своими знаниями. Если среда изначально неизвестна, агенту необходимо изучить ее. В этом случае агент должен не только собирать информацию об окружающей среде, но и учиться на основе полученных данных. Прикладной программный агент может обладать некоторыми предварительными знаниями об окружающей среде, но эти знания могут изменяться и дополняться по мере того, как агент приобретает опыт.

Прикладные программные агенты МАИС сотрудничают для оптимизации общей долгосрочной цели. Для адаптации к нестационарным условиям среды и поиска оптимальных стратегий в МАИС используются алгоритмы многоагентного обучения с подкреплением. В МАИС могут использоваться как алгоритмы обучения с подкреплением на основе модели среды, так и безмодельные алгоритмы.

В алгоритмах первого типа используются точные сведения об окружающей среде, в которой работают программные агенты. Как и KBS-системы, МАИС, использующие MARL на основе моделей, обучаются эффективно, т.к. не тратят время на поиск неправильных путей. В случае, если среда имеет строгие статические правила, для повышения эффективности МАИС необходимо разработать индивидуальный MARL-алгоритм на основе модели. Если же алгоритм обучения с подкреплением должен быть применим к любой промышленной задаче, то используются безмодельные алгоритмы, согласно которым программные агенты изучают стратегии через взаимодействия, усваивая при этом любые правила окружающей среды. Наиболее продвинутые алгоритмы обучения с подкреплением позволяют одновременно с изучением стратегий изучать модели среды и использовать потенциальные, но неизвестные действия других агентов.

РАСПРЕДЕЛЕННАЯ БАЗА ЗНАНИЙ МАИС

Для представления и обработки знаний в МАИС может использоваться описанная авторами в работах [1, 2] интеллектуальная подсистема управления РБЗ. Для расширения возможностей этой подсистемы была осуществлена трансформация когнитивного

ядра прикладных программных агентов путем включения LLM и векторной базы знаний в состав МАИС.

Большая языковая модель развернута локально и работает в реальном времени при возможных ограниченных ресурсах. Векторная база знаний реализована как надстройка над РБЗ и используется для семантического поиска, ускоряющего и расширяющего логический вывод (рис. 3).

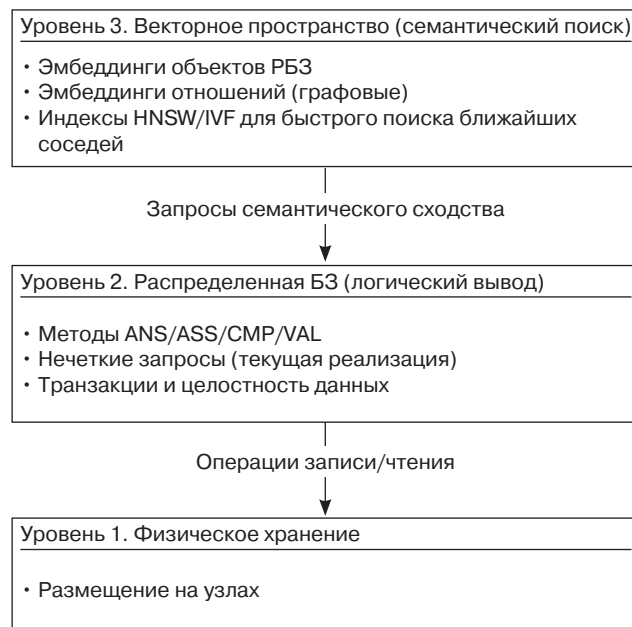


Рис. 3. Векторная база знаний как расширение РБЗ. HNSW (Hierarchical Navigable Small World) – алгоритм приближенного поиска ближайших соседей, основанный на слоистой графовой структуре данных; IVF (Inverted File Index) – алгоритм приближенного поиска ближайших соседей, использующий кластеризацию данных и ограничение области поиска

Преобразование объектов базы знаний в векторное пространство осуществляется через модели типа text-embedding-3-large¹ для векторизации текстовых атрибутов, графовые эмбединги для кодирования реляционных связей. Использование LLM, моделей text-embedding-3-large и графовых эмбедингов позволяет создать унифицированное семантическое пространство, в котором агенты способны выполнять операции поиска смысловых близостей, обобщения и ассоциативного вывода. Это существенно расширяет когнитивные возможности агентов, обеспечивая более глубокое понимание контекста задач и повышая эффективность коллективного взаимодействия. Дальнейшее развитие подхода видится

¹ Модель встраивания текста (эмбединга), предназначенная для преобразования естественного языка в высокоточные числовые векторы, которые сохраняют семантическое значение. [A text embedding model designed to convert natural language into high-precision numeric vectors that preserve semantic meaning.]

в исследовании методов динамического обновления векторных представлений в процессе функционирования системы и разработке механизмов интерпретации результатов семантического поиска для объяснения принимаемых агентами решений² [13, 14].

ЭКСПЕРИМЕНТАЛЬНАЯ ВАЛИДАЦИЯ И АНАЛИЗ ЭФФЕКТИВНОСТИ МАИС

Для подтверждения практической значимости разработанной архитектуры МАИС и обоснования эффективности заложенных в нее механизмов проведен комплексный анализ, включающий экспериментальное моделирование ключевых подсистем. Валидация опиралась на прототипирование отдельных компонентов и анализ их поведения в контролируемых сценариях, соответствующих условиям неполноты данных и динамичности среды.

Для оценки эффективности координации прикладных программных агентов в условиях нестационарной среды проведено сравнительное исследование двух репрезентативных алгоритмов MARL, применимых в рамках предложенной децентрализованной архитектуры: алгоритма на основе смешивания Q-значений (Q-value Mixing Network, QMIX) и алгоритма оптимизации проксимальной политики (Multi-Agent Proximal Policy Optimization, MAPPO). Эксперимент проводился на наборе сценариев, моделирующих кооперативное взаимодействие прикладных программных агентов при решении распределенной задачи диагностики. Полученные результаты демонстрируют, что MAPPO обеспечивает более стабильное обучение и лучшую сходимость. Несмотря на высокую эффективность, QMIX демонстрирует чувствительность к монотонности функции смешивания, что ограничивает его применимость в сценариях со сложной структурой кооперации.

Для оценки эффективности интеграции LLM и векторной базы знаний был реализован прототип подсистемы семантического поиска. В качестве основы использовалась связка из локально развернутой LLM и векторной базы данных, интегрированной с РБЗ. Эксперимент проводился на наборе данных, включающем прецеденты из проблемной области диагностики сложных технических систем. Результаты моделирования показали, что по сравнению с базовой архитектурой МАИС, в которой доступ к знаниям осуществляется только через SQL-запросы и продукционные правила, гибридная архитектура обеспечивает более полный семантический поиск при незначительном снижении точности, что компенсируется ранжированием результатов по степени релевантности.

Полученные экспериментальные данные свидетельствуют, что предложенная архитектура МАИС, основанная на кооперации прикладных и системных программных агентов, обеспечивает высокую эффективность при решении сложных задач в динамически изменяющихся средах. Ключевыми факторами выступают: устойчивая координация на основе MAPPO, расширение когнитивных возможностей агентов за счет LLM и векторной базы знаний, а также адаптивное управление ресурсами с помощью системных агентов-диспетчеров.

ЗАКЛЮЧЕНИЕ

Проведенное исследование обосновывает целесообразность и концептуальную завершенность агентно-ориентированного подхода к построению распределенных интеллектуальных систем, функционирующих в условиях неполноты исходных данных, высокой стохастичности и нестационарности внешней среды. Представленная в работе архитектура МАИС интегрирует две основные парадигмы искусственного интеллекта: символичные методы представления и обработки знаний и нейросетевые алгоритмы. Такая гибридная когнитивная архитектура наделяет программных агентов способностью не только адаптироваться к динамическим изменениям среды, но и сохранять рациональность поведения в условиях, где чистые нейросетевые подходы или классические экспертные системы демонстрируют несостоятельность. Агентно-ориентированная парадигма является не просто одной из альтернатив, а наиболее адекватной методологической основой для построения распределенных интеллектуальных систем, предназначенных для слабоформализуемых проблемных областей. В работе представлено обоснование инженерно-технических решений по реализации агентно-ориентированных систем на современных вычислительных платформах, включая взаимодействие с микросервисной инфраструктурой, организацию межпроцессных коммуникаций и балансировку вычислительной нагрузки. Разработанная теоретическая и технологическая база открывает перспективы для практической реализации эффективных и надежных распределенных интеллектуальных систем. Дальнейшие исследования целесообразно направить на углубление интеграции больших языковых моделей в когнитивное ядро агентов, что позволит обогатить семантику межсубъектного взаимодействия и повысить качество интерпретации слабоструктурированной информации.

Вклад авторов

Все авторы в равной степени внесли свой вклад в исследовательскую работу.

Authors' contribution

All authors contributed equally to the research work.

² HUN-REN SZTAKI. *Distributed Knowledge Graphs (DKG): Architecture and Deployment Patterns*. COST Action CA20136. Technical Report. Budapest. 2024. 48 p.

СПИСОК ЛИТЕРАТУРЫ / REFERENCES

1. Зайцев Е.И., Нурматова Е.В. О подходе к управлению знаниями и разработке мультиагентной системы представления и обработки знаний. *Russian Technological Journal*. 2023;11(4):16–25. <https://doi.org/10.32362/2500-316X-2023-11-4-16-25> [Zaytsev E.I., Nurmatova E.V. Approach to knowledge management and the development of a multi-agent knowledge representation and processing system. *Russian Technological Journal*. 2023;11(4):16–25. <https://doi.org/10.32362/2500-316X-2023-11-4-16-25>]
2. Zaytsev E.I., Nurmatova E.V. Increasing the performance of a multi-agent knowledge representation and processing system. *Phys. Part. Nuclei*. 2024;55:384–386. <https://doi.org/10.1134/S1063779624030948>
3. Nguyen T.T., Nguyen N.D., Nahavandi S. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Transactions on Cybernetics*. 2020;50(9):3826–3839. <https://doi.org/10.1109/TCYB.2020.2977374>
4. Zhang K., Yang Z., Başar T. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. In: Vamvoudakis K.G., Wan Y., Lewis F.L., Canseve D. (Eds.). *Handbook of Reinforcement Learning and Control. Studies in Systems, Decision and Control*. Springer; 2021. V. 325. P. 321–384. https://doi.org/10.1007/978-3-030-60990-0_12
5. Min Y., He J., Wang T., Gu Q. Cooperative Multi-Agent Reinforcement Learning: Asynchronous Communication and Linear Function Approximation. In: *ICML '23: Proceedings of the 40th International Conference on Machine Learning*. 2023. P. 24785–24811. *arXiv preprint*. [arXiv:2305.06446](https://arxiv.org/abs/2305.06446). <https://doi.org/10.48550/arXiv.2305.06446>
6. Vasilev B., Gupta T., Peng B., Whiteson S. Semi-On-Policy Training for Sample Efficient Multi-Agent Policy Gradients. *arXiv preprint*. [arXiv:2104.13446](https://arxiv.org/abs/2104.13446). 2021. <https://doi.org/10.48550/arXiv.2104.13446>
7. Wang H., Liu Z., Yi J., Pu Z. Multiagent hierarchical cognition difference policy for multiagent cooperation. *Algorithms*. 2021;14(3):Article 98. <https://doi.org/10.3390/a14030098>
8. Zeng S., Doan T.T., Romberg J. A two-time-scale stochastic optimization framework with applications in control and reinforcement learning. *SIAM J. Optimization*. 2024;34(1):946–976. <https://doi.org/10.1137/22M150277X>
9. Guo T., Chen X., Wang Y., Chang R., Pei S., Chawla N.V., Wiest O., Zhang X. Large Language Model based Multi-Agents: A Survey of Progress and Challenges. 2024. *arXiv preprint*. [arXiv:2402.01680](https://arxiv.org/abs/2402.01680). <https://doi.org/10.48550/arXiv.2402.01680>
10. Lim J., Vogel-Heuser B., Kovalenko I. Large Language Model-Enabled Multi-Agent Manufacturing Systems. *arXiv preprint*. [arXiv:2406.01893](https://arxiv.org/abs/2406.01893). 2024. <https://doi.org/10.48550/arXiv.2406.01893>
11. Zhang Y., Sun R., Chen Y., Pfister T., Zhang R., Arik S. Chain of Agents: Large Language Models Collaborating on Long-Context Tasks. In: *Advances in Neural Information Processing Systems*. 2024;37. December 10–15, 2024. Vancouver, Canada. <https://doi.org/10.52202/079017-4202>
12. Yang Y., Chai H., Shao S., Song Y., Qi S., Rui R., Zhang W. AgentNet: Decentralized Evolutionary Coordination for LLM-based Multi-Agent Systems. *arXiv preprint*. [arXiv:2504.00587](https://arxiv.org/abs/2504.00587). 2025. <https://doi.org/10.48550/arXiv.2504.00587>
13. Bai J., Mosbach S., Taylor C.J., Karan D., Lee K.F., Rihm S.D., Akroyd J., Lapkin A.A., Kraft M. A Dynamic Knowledge Graph Approach to Distributed Self-Driving Laboratories. *Nat. Commun.* 2024;15(1):Article 462. <https://doi.org/10.1038/s41467-023-44599-9>
14. Zhou W., Yan Y., Yang Q. DGRAG: Distributed Graph-based Retrieval-Augmented Generation for Edge-Cloud Systems. *arXiv preprint*. [arXiv:2505.19847](https://arxiv.org/abs/2505.19847). 2025. <https://doi.org/10.48550/arXiv.2505.19847>

Об авторах

Зайцев Евгений Игоревич, к.т.н., доцент, кафедра КБ-14 «Цифровые технологии обработки данных», Институт кибербезопасности и цифровых технологий, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: zajcev@mirea.ru. Scopus Author ID 57218190023, ResearcherID ABA-4823-2020, SPIN-код РИНЦ 9662-7658, <https://orcid.org/0000-0002-1979-5611>

Нурматова Елена Вячеславовна, к.т.н., доцент, кафедра КБ-3 «Разработка программных решений и системное программирование», Институт кибербезопасности и цифровых технологий, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: nurmatova@mirea.ru. Scopus Author ID 57205460003, ResearcherID GQI-3212-2022, SPIN-код РИНЦ 7036-3661, <https://orcid.org/0000-0001-8511-0978>

About the Authors

Evgeniy I. Zaytsev, Cand. Sci. (Eng.), Associate Professor, Department of Digital Data Processing Technologies, Institute for Cybersecurity and Digital Technologies, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: zajcev@mirea.ru. Scopus Author ID 57218190023, ResearcherID ABA-4823-2020, RSCI SPIN-code 9662-7658, <https://orcid.org/0000-0002-1979-5611>

Elena V. Nurmatova, Cand. Sci. (Eng.), Associate Professor, Department of Software Development and System Programming, Institute for Cybersecurity and Digital Technologies, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: nurmatova@mirea.ru. Scopus Author ID 57205460003, ResearcherID GQI-3212-2022, RSCI SPIN-code 7036-3661, <https://orcid.org/0000-0001-8511-0978>