

ПРИМЕНЕНИЕ МУЛЬТИАГЕНТНЫХ ТЕХНОЛОГИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ В ТРАНСПОРТНОЙ ЛОГИСТИКЕ

КОЛОМБЕТ Валерий Андреевич

ассистент кафедры «Корпоративные информационные системы»

РАХМЕТОВА Лидия Евгеньевна

ассистент кафедры «Корпоративные информационные системы»

СТАЩАК Светлана Васильевна

ассистент кафедры «Корпоративные информационные системы»

Севастопольский государственный университет

г. Севастополь, Россия

В данной работе автором поднимается вопрос о целесообразности применения мультиагентных технологий для решения задач транспортной логистики. Рассматриваются характеристики для различных архитектур мультиагентных систем, с учетом их положительных и отрицательных свойств. В статье рассматриваются аспекты применения мультиагентных систем и примеры их реализации.

Ключевые слова: мультиагентные системы (MAS), транспортная логистика, ограничения MAS, архитектуры мультиагентных систем.

Современные логистические и транспортные системы сталкиваются с необходимостью обеспечивать высокую адаптивность, устойчивость и масштабируемость в условиях динамически изменяющейся среды. Классические централизованные архитектуры (система планирования ресурсов (Enterprise Resource Planning – ERP) и система управления транспортными перевозками (Transport Management System – TMS)) обеспечивают глобальную оптимальность за счёт полной предварительной проработки маршрутов и графиков, однако они оказываются малопригодными для работы в условиях неопределенности и

изменчивости, поскольку при малейших изменениях условий, требуют перерасчёта всех затрагиваемых параметров системы. Гибридные архитектуры частично решают проблему масштабируемости, но сохраняют сложности интеграции и централизованного управления.

По причине централизованного подхода ТМС и отсутствия гибкости алгоритмов систем ERP, за последнее десятилетие наблюдается растущий интерес к применению мультиагентных систем во многих областях, в том числе в транспортной логистике. При этом первые исследования мультиагентных систем велись еще с 1980-х годов, но плоды этих исследований появились лишь через несколько десятков лет. Но так ли хороши мультиагентные системы (MAS)? Насколько велика польза от их внедрения?

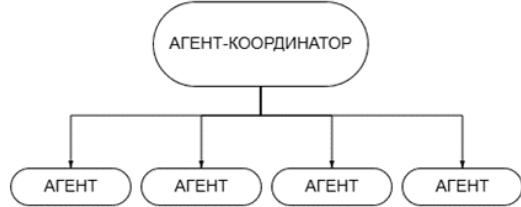
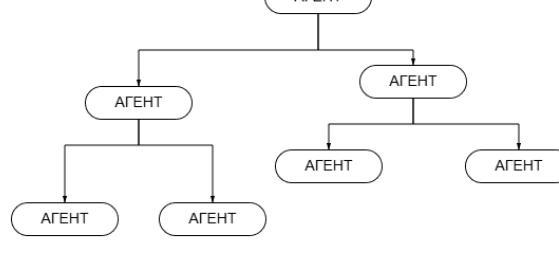
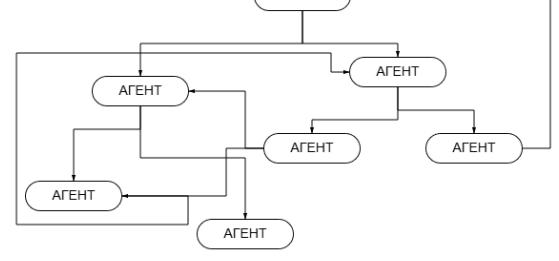
В рамках данной статьи рассматриваются архитектуры мультиагентных систем, их положительные и отрицательные стороны, а также приводятся примеры функционирующих систем, основанных на различных архитектурах.

Мультиагентная система – это совокупность автономных агентов, взаимодействующих для решения общей задачи. Агент определяется как автономный программный модуль с «социальными» навыками взаимодействия. Такие системы хорошо подходят для сложных распределённых задач: они способны учитывать множество автономных объектов с частично разделяемыми данными и локальными целями. Например, цепочка поставок состоит из множества участников (склады, перевозчики, потребители), что естественно моделируется сетью агентов [4]. По сравнению с централизованной системой (единий контроллер), мультиагентная система обеспечивает децентрализацию и гибкость: агенты сами принимают решения локально, обмен информацией позволяет быстро адаптироваться к изменениям ситуации. Важная особенность – появление *эмерджентного поведения* при взаимодействии агентов, что может привести к непредвиденным глобальным эффектам.

Классификация архитектур мультиагентных систем по уровням управления приведена в таблице 1 [1].

Таблица 1.

КЛАССИФИКАЦИЯ АРХИТЕКТУР МУЛЬТИАГЕНТНЫХ СИСТЕМ

Архитектура	Схема архитектуры
<p>Централизованная: существует главный координатор или брокер, а остальные агенты подчинены ему. Подход прост для реализации, но в случае выхода из строя агента-координатора – вся система становится неработоспособной.</p>	 <pre> graph TD K[АГЕНТ-КООРДИНАТОР] --> A1[АГЕНТ] K --> A2[АГЕНТ] K --> A3[АГЕНТ] K --> A4[АГЕНТ] </pre>
<p>Пример. Система вызова такси (такая как Uber или Maxim). Координатором в данной системе выступает сервер, который получает заказы от клиентов, анализирует местоположение свободных водителей (агентов) и определяет кто из них возьмет конкретный заказ. При этом сам водитель не принимает участие в выборе исполнителя заказа – он исполняет назначенную ему задачу – довезти клиента до места назначения.</p>	
<p>Иерархическая (древовидная): агенты распределены по различным уровням управления, образуя древовидную систему подчинения. На каждом уровне находятся агенты с определённым набором функций и полномочий, где вышестоящие агенты контролируют и координируют деятельность подчинённых, а нижестоящие агенты выполняют конкретные задачи. Взаимодействие между уровнями осуществляется вертикально, с передачей команд сверху вниз и обратной связью снизу вверх.</p>	 <pre> graph TD L1[АГЕНТ] --> L2_1[АГЕНТ] L1 --> L2_2[АГЕНТ] L2_1 --> L3_1[АГЕНТ] L2_1 --> L3_2[АГЕНТ] L2_2 --> L3_3[АГЕНТ] L2_2 --> L3_4[АГЕНТ] </pre>
<p>Пример: система автоматизированной логистики на складах Amazon Robotics (бывш. Kiva Systems). В этой архитектуре управление разделено по уровням. На верхнем уровне работают управляющие контроллеры, которые следят за общей логикой распределения заказов и маршрутов, а на нижнем уровне находятся мобильные роботы-агенты, которые перемещают стеллажи и грузы.</p>	
<p>Гетероархическая (слабая иерархия): нет жёсткой иерархии, Агенты могут находиться на разных уровнях иерархии, однако руководящая функция не закреплена строго за конкретным агентом — в зависимости от обстоятельств любой агент может взять на себя координационную роль в системе, если он обладает максимальной полнотой информации или ресурсами для принятия решений в конкретно взятой ситуации.</p>	 <pre> graph TD A1[АГЕНТ] --- A2[АГЕНТ] A1 --- A3[АГЕНТ] A2 --- A4[АГЕНТ] A2 --- A5[АГЕНТ] A3 --- A6[АГЕНТ] A3 --- A7[АГЕНТ] A4 --- A8[АГЕНТ] A5 --- A8 A6 --- A9[АГЕНТ] A7 --- A9 </pre>
<p>Пример: В системах кооперативного автономного транспорта, особенно при организации дорожного движения без светофоров, часто применяется гетероархическая архитектура</p>	

Архитектура	Схема архитектуры
Автомобили (агенты) обмениваются данными напрямую друг с другом и принимают локальные решения. На перекрёстках или в зонах интенсивного трафика временным координатором может стать ближайший к центру ситуации автомобиль, который управляет проездом других. Если такой координатор уезжает или выходит из строя, другой агент принимает на себя руководство. Таким образом, система адаптируется под изменяющиеся условия и сохраняет устойчивость, не нуждаясь в централизованном управляющем элементе.	
Распределённая (пикинговая): все агенты обладают одинаковыми правами а центральный управляющий агент отсутствует. Каждый агент самостоятельно принимает решения, опираясь на локальную информацию и обмен сообщениями с соседними агентами. Такая модель отличается высокой устойчивостью к сбоям и способна сохранять работоспособность даже при выходе из строя отдельных компонентов, однако выработка корректного решения между большим числом агентов может быть затруднена, особенно в быстро меняющейся среде.	<pre> graph TD A([АГЕНТ]) <--> B([АГЕНТ]) A <--> C([АГЕНТ]) A <--> D([АГЕНТ]) B <--> C B <--> D C <--> D </pre>
Рой дронов является наиболее точным примером такой архитектуры, например, поисково-спасательные операции в разрушенных зданиях или мониторинг лесных пожаров. При этом каждый дрон действует независимо, но поддерживает прямую связь с соседними дронами. В ходе работы системы дроны обмениваются данными о местоположении, угрозах, зонах сканирования, чтобы не дублировать работу и распределять участки между собой, а при выходе из строя одного дрона, остальные перераспределяют задачи между собой автоматически, без необходимости внешнего вмешательства.	

Классический централизованный подход в логистике опирается на единую платформу управления и планирования, такую как ERP-системы или централизованные TMS. В таких решениях оптимизация маршрутов и расписаний происходит заранее, с использованием глобальных алгоритмов. Это позволяет достичь высокой общей эффективности, однако система оказывается маломасштабируемой и недостаточно гибкой: даже небольшое изменение в транспортной сети требует полного пересчёта плана движения. К тому же плотные взаимосвязи между компонентами делают всю систему уязвимой при возникновении сбоев. Частично эту проблему помогают решить гибридные архитектуры, например иерархические модели, которые распределяют вычислительные задачи по

уровням, но при этом сталкиваются с трудностями интеграции и координации между различными слоями управления.

В отличии от централизованных решений, мультиагентные системы опираются на принцип локальной автономии и децентрализованного взаимодействия. Поскольку транспортные сети по своей природе являются распределёнными, применение подхода «снизу вверх» значительно облегчает процесс их моделирования и управления. Каждый агент имеет доступ лишь к ограниченной части окружающей среды и действует самостоятельно, согласовывая свои действия с ближайшими участниками. Это обеспечивает гибкое и адаптивное поведение системы в целом. Так, например, в системах управления дорожным движением, каждый перекрёсток может быть оснащён агентом, который самостоятельно переключает сигнальные фазы в зависимости от текущей транспортной обстановки [3]. При этом с помощью обмена данными между перекрёстками формируется согласованная стратегия движения по всей сети. В отличие от этого, централизованное управление требует предварительной обработки полной информации и, как следствие, более медленно реагирует на изменения в дорожной ситуации. В то же время централизованная система регулирования вынуждена ждать обработки всех данных и занимает больше времени на обновление плана движения.

Анализ показывает, что мультиагентные системы обладают значительно большей гибкостью и способностью адаптироваться, особенно в условиях нестабильной или быстро меняющейся среды [3]. Такие системы способны оперативно перераспределять обязанности между агентами, переназначать ресурсы и находить адекватные решения ситуаций при возникновении сбоев или неожиданных событий. В отличие от этого, традиционные подходы демонстрируют высокую эффективность, когда требуется оптимальное решение в задаче с заранее известными исходными условиями, не меняющимися в процессе функционирования системы, но теряют эту эффективность с изменением исходных условий в процессе как только появляются отклонения от исходных условий.

Можно сделать вывод, что мультиагентные решения превосходят традиционные в надёжности и способности к масштабированию, в то время как классические системы требуют значительных ресурсов при расширении и нередко содержат узлы, отказ которых ведет к отказу всей системы целиком.

Несмотря на то, что все приведенные примеры относятся к зарубежным компаниям, в Российской Федерации проводится активное исследование возможностей мультиагентных систем. Особенный интерес представляют проекты в сфере экстренной логистики. Например, в 2023 году в статье [3] была описана концепция мультиагентной системы для диспетчеризации бригад «скорой помощи» в Санкт-Петербурге. Предложенная система включает агентов, которые мониторят положение автомобилей и взаимодействуют с городской транспортной сетью (сигналами светофоров) для увеличения пропускной способности маршрутов «скорой». В статье описаны основные принципы функционирования: агенты получают данные с устройств GPS и ITS, координируются с трафик-контролем, чтобы обнаружить минимальный по времени маршрут для проезда машин экстренных служб. Авторы подчёркивают, что на данный момент речь идёт скорее о теоретическом обосновании, поскольку «состояние финансирования здравоохранения в России не позволяет говорить о немедленном внедрении автономной системы» [2]. Этот пример демонстрирует потенциал мультиагентных систем в локальных логистических задачах: при дальнейшей разработке такой подход может повысить оперативность экстренных служб.

Возможности применения мультиагентных систем практически безграничны. На основе технологий роя дронов во взаимодействии с возможностями мультиагентных систем можно представить систему перевозки посылок (например лекарств) в труднодоступные места Земли – горы Кавказа, Непала и т. д. Например создать дирижабль (брокер для централизованной архитектуры) как центральный логистический узел, который патрулирует регион или зависит над определённой территорией. Из него дроны-агенты запускаются для выполнения заранее определенных задач доставки. Систему можно реализовать в виде

гибридной мультиагентной архитектуры, где дирижабль выполняет функции центра координации, но дроны работают в полуровой, гетероархической среде, координируясь друг с другом в полёте для устранения последствий в виде потери посылок из-за отказа дронов.

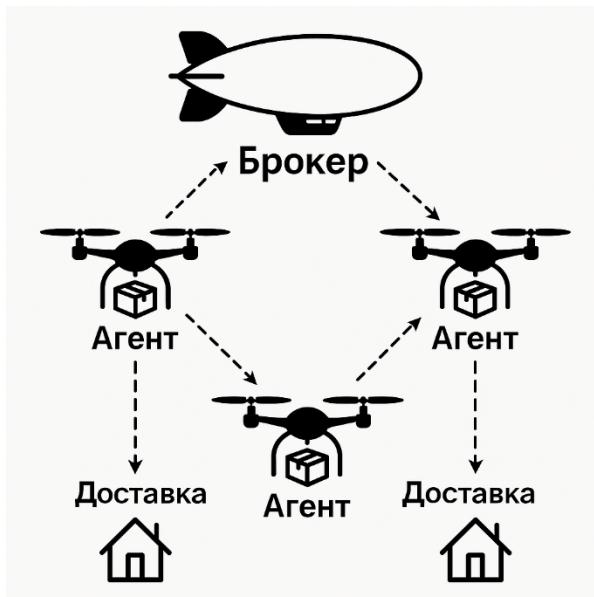


Рисунок 1. Концептуальная схема мультиагентной модели доставки грузов.

Однако всё ли так замечательно в мультиагентных системах? Несмотря на все положительные свойства, у мультиагентных систем имеются ограничения.

Во-первых, скрытое взаимодействие агентов может приводить к непредсказуемым последствиям: отсутствие глобального планировщика требует тщательной валидации сходимости в рамках всего процесса.

Во-вторых, коммуникационные затраты. Распределённость требует обмена сообщениями, что может стать узким местом при большом числе агентов и объёме данных (включая трафик сенсоров интернета вещей).

В-третьих, качественные данные. Эффективность мультиагентных систем сильно зависит от достоверности входной информации (о состоянии дорог, наличии ресурсов, спросе), а данные интернета вещей и цифровых двойников могут быть неполными или запаздывающими, что снижает точность решений.

Это объясняет факт того, что на сегодня мультиагентные решения в логистике часто остаются в стадии пилотных испытаний и моделирования, а их промышленная реализация требует дополнительных усилий по верификации, тестированию и адаптации под конкретные условия.

Анализ мультиагентных систем, применительно к задачам транспортной логистики показывает, что перспективным направлением развития является создание гибридных архитектур, сочетающих централизованное стратегическое планирование с агентной тактической адаптацией. При этом несмотря на очевидные преимущества MAS по гибкости и устойчивости, остаются открытыми вопросы о сопоставимой эффективности таких систем в задачах логистики по сравнению с традиционными архитектурами.

ЛИТЕРАТУРА

1. An A, Wang H., Huang Y., Parlos A. G. Multi-agent system architectures for collaborative prognostics // Journal of Intelligent Manufacturing. – 2019. – Vol. 30. – P. 2999–3013. – DOI: 10.1007/s10845-019-01478-9. – URL: <https://link.springer.com/article/10.1007/s10845-019-01478-9> (дата обращения: 03.05.2025).
2. Ilyashenko V., Lukyanchenko E., Zheleznyak A., Likhita N. Implementing Multi-agent System for Dispatch and Control of Emergency Medical Care Ambulances to Enhance Emergency Response Effectiveness and Quality in Saint-Petersburg // Lecture Notes on Data Engineering and Communications Technologies. – 2023. – T. 157. – C. 221–234. – DOI: 10.1007/978-3-031-24434-6_18.
3. Lijun Sun, Yijun Yang, Qiqi Duan, Yuhui Shi, Chao Lyu, Yu-Cheng Chang, Chin-Teng Lin, Yang Shen Multi-Agent Coordination across Diverse Applications: A Survey [Электронный ресурс] // arXiv:2502.14743. – 2024. – URL: <https://arxiv.org/abs/2502.14743> (дата обращения: 09.05.2025).
4. Xu L., Mak S., Minaricova M., Brintrup A. On Implementing Autonomous Supply Chains: a Multi-Agent System Approach [Электронный ресурс] //

arXiv:2310.09435. – 2023. – URL: <https://arxiv.org/abs/2310.09435> (дата обращения: 20.04.2025).

APPLICATION OF MULTI-AGENT TECHNOLOGIES FOR SOLVING PROBLEMS IN TRANSPORT LOGISTICS

KOLOMBET Valeriy Andreevich

Assistant of the Department of Corporate Information Systems

RAKHMETOVA Lidiya Evgenyevna

Assistant of the Department of Corporate Information Systems

STASHCHAK Svetlana Vasilievna

Assistant of the Department of Corporate Information Systems

Sevastopol State University

Sevastopol, Russia

This paper addresses the feasibility of using multi-agent technologies to solve transportation logistics problems. The author examines the characteristics of various multi-agent system architectures, considering their advantages and disadvantages. The article discusses aspects of applying multi-agent systems and provides examples of their implementation.

Keywords: multi-agent systems (MAS), transportation logistics, MAS limitations, multi-agent system architectures.