

АРХИТЕКТУРА МНОГОАГЕНТНОЙ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ПАРАМЕТРИЧЕСКОЙ МОДЕЛИ ДЛЯ АНАЛИЗА РАСПРЕДЕЛЕННЫХ ПРОГРАММНО-АППАРАТНЫХ КОМПЛЕКСОВ**Р.Г. Астраханцев, С.П. Бобков**

Роман Геннадьевич Астраханцев (ORCID 0000-0001-9880-2826)*,
Сергей Петрович Бобков (ORCID 0000-0001-7315-1625)
Ивановский государственный химико-технологический университет,
Россия. 153000, Ивановская область, г. Иваново, Шереметевский пр-т, д.7
E-mail rgastrakhantsev@gmail.com*, bsp@isuct.ru

Разработана параметрическая n-уровневая иерархическая архитектура много-агентной интеллектуальной системы для автоматизированного анализа состояния распределенных программно-аппаратных комплексов. Предложено авторское определение многоагентной интеллектуальной системы, расширяющее ГОСТ Р 59277–2020 за счет фиксации стохастических условий функционирования и частичной неопределенности среды. На основе теории иерархических многоуровневых систем и теории управления организационными системами обоснован выбор трехуровневой конфигурации, включающей стратегический уровень (агент-координатор), тактический уровень (агенты-тимлиды) и операционный уровень (агенты-исполнители). Формализованы функции каждого уровня и введена конфигурация агента как кортеж системной инструкции и набора доступных инструментов при использовании единой языковой модели. Предложена гибридная схема координации, объединяющая четыре механизма информационного обмена между агентами. Введено понятие графа делегирования как эмерджентной структуры, формируемой решениями агентов в процессе функционирования системы. Проведен сравнительный анализ предлагаемой архитектуры с альтернативными подходами к организации многоагентных систем.

Ключевые слова: многоагентная интеллектуальная система, иерархическая архитектура, распределенный программно-аппаратный комплекс, координация агентов, граф делегирования, пространство артефактов, системный анализ

PARAMETRIC MODEL-BASED ARCHITECTURE OF A MULTI-AGENT INTELLIGENT SYSTEM FOR DISTRIBUTED COMPUTING ENVIRONMENT ANALYSIS**R.G. Astrakhantsev, S.P. Bobkov**

Roman G. Astrakhantsev (ORCID 0000-0001-9880-2826)*,
Sergey P. Bobkov (ORCID 0000-0001-7315-1625)
Ivanovo State University of Chemistry and Technology,
Russia. 153000, Ivanovo region, Ivanovo, Sheremetevsky ave., 7
E-mail rgastrakhantsev@gmail.com*, bsp@isuct.ru

A parametric n-level hierarchical architecture model of a multi-agent intelligent system for automated state analysis of distributed software-hardware complexes is developed. The authors propose a definition of a multi-agent intelligent system that extends GOST R 59277–2020 by incorporating stochastic operating conditions and partial environmental uncertainty. Based on the theory of hierarchical multilevel systems and the theory of organizational systems management, a three-level configuration is justified, comprising a strategic level (coordinator agent), a tactical level (team leader agents), and an operational level (executor agents). Functions of each

level are formalized, and agent configuration is defined as a tuple of system instruction and available tool set under a single language model. A hybrid coordination model combining four information exchange mechanisms is proposed. The concept of a delegation graph as an emergent structure formed by agent decisions during system operation is introduced. A comparative analysis of the proposed architecture against alternative approaches to multi-agent system organization is conducted.

Keywords: multi-agent intelligent system, hierarchical architecture, distributed software-hardware complex, agent coordination, delegation graph, artifact space, systems analysis

Для цитирования:

Астраханцев Р.Г., Бобков С.П. Архитектура многоагентной интеллектуальной системы на основе параметрической модели для анализа распределенных программно-аппаратных комплексов. *Современные наукоёмкие технологии. Региональное приложение*. 2026. № 2. С. 70-81. DOI: 10.6060/snt.20268602.08

For citation:

Astrakhantsev R.G., Bobkov S.P. Architecture of a multi-agent intelligent system based on a parametric model for the analysis of distributed software and hardware systems. *Modern high technology. Regional application*. 2026. N 2. P. 70-81. DOI: 10.6060/snt.20268602.08

ВВЕДЕНИЕ

Задачи автоматизированного анализа состояния распределенных программно-аппаратных комплексов требуют одновременной обработки разнородной информации, координации множества параллельных процессов и принятия решений в условиях неполноты данных [1, 2]. Существующие подходы опираются преимущественно на одноагентные системы или плоские многоагентные архитектуры с заранее определенным составом агентов [3, 4]. Ни те ни другие не обеспечивают масштабируемой декомпозиции задач при росте сложности исследуемого комплекса.

Одноагентные системы сосредоточивают все функции в единственном программном модуле, что ограничивает возможности параллельного исследования и вынуждает один и тот же компонент решать задачи принципиально разного масштаба [5, 6]. Плоские многоагентные архитектуры распределяют функции между несколькими равноправными агентами, однако отсутствие иерархии затрудняет координацию при увеличении числа участников, поскольку каждый агент вынужден поддерживать прямое взаимодействие со всеми остальными [4, 7]. Специализация агентов в подобных системах, как правило, задается при проектировании и не может изменяться в процессе работы, что исключает адаптацию к непредвиденным особенностям исследуемого комплекса.

Альтернативой является иерархическая декомпозиция, позволяющая разделить задачу на подзадачи различного масштаба и распределить их между уровнями с различной степенью абстракции [8, 9]. Теория иерархических много-

уровневых систем Месаровича, Мако и Такахары [10] устанавливает принципы построения таких структур и связывает число уровней с глубиной естественной декомпозиции решаемой задачи. Теория управления организационными системами Буркова и Новикова [11] развивает эти идеи применительно к координации автономных элементов с собственными критериями эффективности. Вместе с тем вопросы применения данных теоретических оснований к проектированию многоагентных интеллектуальных систем с динамическим формированием состава агентов остаются недостаточно проработанными.

Авторами ранее разработана формальная модель задачи автоматизированного анализа на основе расширения аппарата децентрализованного частично наблюдаемого марковского процесса принятия решений [1]. Модель включает комбинаторное пространство состояний, динамическое пространство действий и механизм верификации достижения целевого состояния. Однако она определяет оптимизационную задачу, но не устанавливает архитектуру системы, которая эту задачу решает.

Целью настоящей статьи является разработка иерархической архитектуры многоагентной интеллектуальной системы, определяющей структуру множества агентов, механизмы их координации и принципы информационного обмена. Основные результаты работы состоят в следующем. Предложена параметрическая n -уровневая архитектура с обоснованием выбора трехуровневой конфигурации. Формализованы функции стратегического, тактического и операционного уровней. Разработана гибридная схема координации,

объединяющая четыре механизма информационного обмена. Введено понятие графа делегирования как эмерджентной структуры, отражающей текущую конфигурацию системы.

МАТЕРИАЛЫ И МЕТОДЫ

Теоретической основой разработки архитектуры послужили три группы методов. Теория иерархических многоуровневых систем Месаровича, Мако и Такахары [10] предоставила аппарат для анализа многоуровневых структур управления, обоснования числа уровней декомпозиции и определения принципов координации между ними. Теория управления организационными системами Буркова и Новикова [11] использована для формализации взаимодействия автономных элементов, обладающих собственными критериями принятия решений, и проектирования механизмов информационного обмена. Математический аппарат децентрализованного частично наблюдаемого марковского процесса принятия решений [1, 12–14] обеспечил формальную связь архитектуры с оптимизационной моделью задачи автоматизированного анализа через понятие политики агента [15]. Методы криптографического анализа распределенных систем [16] использованы для верификации подходов к обеспечению безопасности межагентного взаимодействия.

Нормативной основой для определения базовых понятий послужил ГОСТ Р 59277–2020, устанавливающий термины и определения в области многоагентных систем [17]. Для организации семантического поиска в хранилище результатов работы агентов применен аппарат векторных представлений текстовых описаний, позволяющий оценивать степень смысловой близости между записями без точного совпадения ключевых слов [18].

РЕЗУЛЬТАТЫ

Прежде чем перейти к описанию архитектуры необходимо уточнить понятийный аппарат. Авторами предлагается следующее определение. Многоагентная интеллектуальная система представляет собой организованную в виде иерархической или сетевой структуры совокупность автономных программных сущностей, именуемых агентами, обладающих способностью к восприятию и интерпретации внешней среды, принятию решений, целенаправленному действию, а также взаимодействию друг с другом, функционирующих в стохастических условиях с частичной или полной неопределенностью, реализующих процессы адаптации, самообучения и коллективного поведения для достижения индивидуальных и коллективных целей.

Предложенное определение согласуется с ГОСТ Р 59277–2020 [17], где многоагентная система трактуется как система, состоящая из множества взаимодействующих интеллектуальных агентов. Вместе с тем авторское определение шире стандартного в двух отношениях. Оно фиксирует интеллектуальный характер агентов, то есть способность к автономному решению задач и системный эффект от их объединения. Оно также акцентирует стохастические условия и частичную неопределенность, что является критически значимым для прикладных задач анализа распределенных программно-аппаратных комплексов, где среда по своей природе динамична и не полностью наблюдаема.

Проектирование архитектуры опирается на фундаментальные принципы системного анализа [2, 8]. Принцип иерархической декомпозиции позволяет разбить сложную систему на уровни с различной степенью абстракции, где каждый уровень оперирует собственным масштабом задач. Принцип модульности обеспечивает независимость компонентов и возможность их замены без перестройки всей системы. Принцип распределенного управления создает основу для координации без жесткой связности.

Авторами предлагается рассматривать иерархическую архитектуру как параметрическую структуру с n уровнями, где $n \geq 1$. Множество агентов системы I разбивается на n непересекающихся подмножеств в соответствии с уровнем иерархии

$$I = I_1 \cup I_2 \cup \dots \cup I_n, I_k \cap I_l = \emptyset \text{ при } k \neq l, \quad (1)$$

где I_k – множество агентов k -го уровня иерархии при $k = 1, \dots, n$.

Условие пустого пересечения обеспечивает однозначность принадлежности каждого агента определенному уровню. Агенты верхних уровней (с меньшими значениями k) отвечают за формирование целей и планирование, нижних (с большими k) за непосредственное взаимодействие с исследуемой средой.

Поведение агента a_i на любом уровне иерархии определяется его политикой, представляющей собой отображение

$$\pi_i : H_i \rightarrow \Delta(A_i), \quad (2)$$

где H_i – история взаимодействий агента, включающая полученные задачи, выполненные действия, наблюдения и межагентные сообщения; A_i – множество доступных действий агента, определяемое его конфигурацией и уровнем иерархии.

Конкретный состав A_i различается для агентов разных уровней и может включать создание дочерних агентов, назначение задач, вызов

инструментов анализа, отправку межагентных сообщений и формирование отчетов.

Авторами вводится понятие пространства артефактов, под которым понимается разделяемое хранилище описаний фактов, обнаруженных агентами всех уровней иерархии при взаимодействии со средой, обеспечивающее накопление, векторное представление и извлечение записей посредством поиска по семантической близости. Пространство является глобально доступным, и любой агент может записывать в него результаты и извлекать записи других агентов независимо от положения в иерархии.

Авторами вводится такой механизм координации как пространство артефактов, определяемый как кортеж

$$\Psi = \langle J, \varphi, write, search \rangle, \quad (3)$$

где J – множество текстовых описаний фактов, обнаруженных агентами при взаимодействии со средой K ; φ – отображение, сопоставляющее каждому описанию $\varphi: J \rightarrow R^d$ его векторное представление в d -мерном пространстве; $write$ – операция записи; $search$ – операция поиска.

Извлечение осуществляется не по точному совпадению ключей, а по семантической близости запроса к имеющимся записям, что позволяет агенту находить релевантные результаты других агентов без знания точной терминологии, использованной агентом-источником.

Авторами разработано графическое представление общей n -уровневой архитектуры, приведенное на рис. 1.

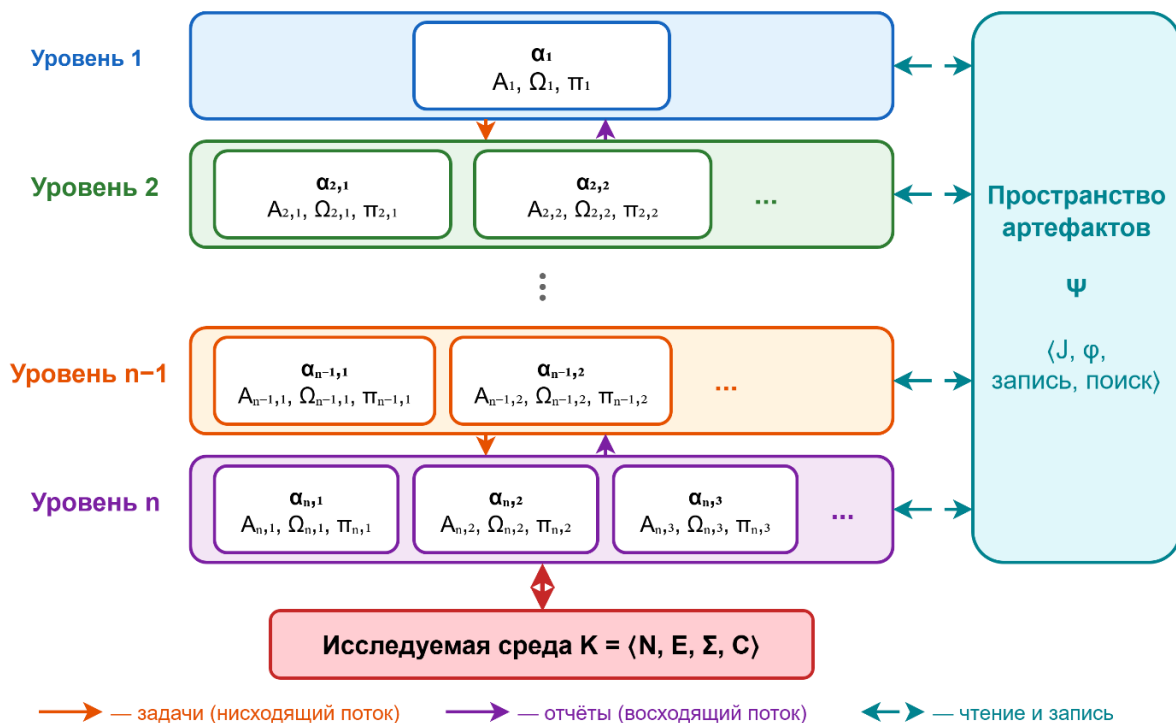


Рис. 1. Общая n -уровневая архитектура многоагентной системы

Fig. 1. General n -level architecture of the multi-agent system

Разработано авторами

При $n = 1$ архитектура вырождается в одноагентную систему, в которой единственный агент совмещает функции планирования и исполнения. При увеличении n каждый дополнительный уровень вводит промежуточную абстракцию между стратегическим планированием и непосредственным исполнением.

Выбор конкретного значения n определяется структурой решаемой задачи. Авторами выделены три масштаба принятия решений при анализе состояния распределенного программно-

аппаратного комплекса. Первый масштаб связан с глобальным целеполаганием, то есть формированием стратегии анализа, определением приоритетов и распределением ресурсов между направлениями исследования. Второй масштаб связан с тактическим управлением, то есть декомпозицией стратегических задач на последовательности конкретных операций, выбором методов и инструментов для каждого направления. Третий масштаб связан с непосредственным исполнением, то есть выполнением атомарных опе-

раций взаимодействия с исследуемым комплексом и получением наблюдений.

Эти три масштаба не сводятся один к другому. Стратегическое решение о приоритете исследования определенного сегмента сети качественно отличается от тактического решения о выборе метода анализа конкретной службы, которое в свою очередь отличается от операционного решения о параметрах конкретного запроса.

Для выбора конкретного значения n проведен анализ альтернативных вариантов. При $n = 2$ агент верхнего уровня вынужден совмещать функции стратегического планирования и тактического управления, что приводит к перегрузке контекста и снижению качества решений на обоих масштабах. При $n = 4$ и более уровнях каждый дополнительный уровень вводит промежуточное звено коммуникации, увеличивающее задержки и потери информации. Трехуровневая декомпозиция является типовой для задач с аналогичным распределением масштабов принятия решений и соответствует канонической форме многоуровневой системы в теории Месаровича, Мако и Такахары [10] с разделением уровней координации, адаптации и исполнения. В теории Буркова и Новикова [11] трехуровневая иерархия также признается минимально достаточной для систем, в которых присутствуют задачи планирования, оперативного управления и непосредственного исполнения.

Авторами предлагается трехуровневая архитектура как частный случай общей n -уровневой системы при $n = 3$

$$I = I_{\text{str}} \cup I_{\text{tac}} \cup I_{\text{oper}}, \quad (4)$$

где I_{str} – множество агентов стратегического уровня; I_{tac} – множество агентов тактического уровня; I_{oper} – множество агентов операционного уровня.

Мощности указанных множеств определяются следующим образом

$$|I_{\text{str}}| = 1, |I_{\text{tac}}| = m(t), |I_{\text{oper}}| = p(t), \quad (5)$$

где $m(t)$ – число агентов тактического уровня в момент времени t ; $p(t)$ – число агентов операционного уровня в момент t .

Стратегический уровень содержит единственного агента для обеспечения согласованности глобальной стратегии. Тактический и операционный уровни формируются динамически в зависимости от текущей задачи и хода анализа.

Стратегический уровень представлен единственным агентом-координатором, реализующим функцию глобального планирования. Формально функция стратегического уровня определяется как

$$f_{\text{str}} : G \times H_{\text{str}} \rightarrow T, \quad (6)$$

где G – множество глобальных целей анализа, задаваемых извне; H_{str} – история взаимодействий

стратегического агента, включающая реестр созданных тактических агентов, назначенные им задачи и полученные отчеты; T – множество тактических задач.

Стратегический агент анализирует поставленную цель, оценивает накопленную информацию об исследуемом комплексе, выявляет направления исследования и формирует набор тактических задач для агентов нижележащего уровня. Координатор не взаимодействует с исследуемым комплексом непосредственно, и вся информация о среде поступает к нему через отчеты тактических агентов.

Тактический уровень состоит из агентов, именуемых тимлидами, каждый из которых специализируется на определенном направлении анализа. Формально функция тактического уровня определяется как

$$f_{\text{tac}} : T \times H_{\text{tac}} \rightarrow A^*, \quad (7)$$

где T – тактическая задача, полученная от стратегического уровня; H_{tac} – история взаимодействий тактического агента с операционным уровнем и накопленные результаты; A^* – последовательность подзадач для операционного уровня.

Тактические агенты создаются стратегическим агентом динамически в соответствии с текущими потребностями. Специализация каждого тимлида определяется координатором в момент создания на основе анализа текущей ситуации.

Операционный уровень состоит из агентов, выполняющих атомарные операции взаимодействия с исследуемым комплексом. Формально функция операционного уровня определяется как

$$f_{\text{oper}} : A \times H_{\text{oper}} \rightarrow O, \quad (8)$$

где A – конкретное действие, назначенное тактическим уровнем; H_{oper} – история взаимодействий операционного агента с инструментами и средой, включающая предыдущие попытки, полученные ошибки и промежуточные результаты; O – наблюдение, полученное в результате выполнения.

Операционные агенты являются единственными агентами в системе, имеющими непосредственный доступ к инструментам анализа. Взаимодействие с инструментами осуществляется через стандартизированный протокольный интерфейс, обеспечивающий унификацию вызовов независимо от конкретного инструмента.

Единая языковая модель используется на всех уровнях иерархии, поскольку навыки, необходимые агентам, по существу совпадают и включают декомпозицию задач, формирование инструкций для дочерних агентов, интерпретацию результатов и координацию. Авторами предлагается формализовать конфигурацию агента a_i как кортеж

$$cfg(a_i) = \langle \psi_i, Tools_i \rangle, \quad (9)$$

где ψ_i – системная инструкция агента, определяющая его роль, правила поведения и границы ответственности; $Tools_i \subseteq Tools_{full}$ – подмножество доступных инструментов из полного каталога.

Обучение на едином корпусе данных позволяет модели накапливать навыки совместно, а

улучшение качества транслируется на все уровни иерархии.

Специализация каждого агента определяется текущей задачей и формируется в момент создания через сочетание системной инструкции и набора инструментов (табл. 1).

Таблица 1

Распределение инструментов по уровням архитектуры
Table 1. Tool distribution across architecture levels

Уровень	Создание дочерних агентов	Назначение задач	Инструменты анализа среды	Пространство артефактов
Стратегический	Да	Да	Нет	Да
Тактический	Да	Да	Нет	Да
Операционный	Нет	Нет	Да	Да

Разработано авторами

Координация в предлагаемой системе осуществляется посредством сочетания иерархической передачи задач и нескольких механизмов информационного обмена. В отличие от архитектур, построенных по паттерну общей доски, где единое разделяемое хранилище является единственным каналом координации, предлагаемая авторами архитектура сочетает локальные механизмы с централизованным пространством артефактов Ψ . Такая гибридная организация позволяет разгрузить централизованное хранилище, ограничив его функцию накоплением фактов для горизонтального обмена, тогда как оперативная координация осуществляется через локальные каналы.

Каждый агент поддерживает собственную локальную память, содержащую историю взаимодействий. Формально

$$H_i = (m_1, m_2, \dots, m_k), \quad (10)$$

где m_k – элемент истории, который может представлять собой задачу, наблюдение, результат действия или межагентное сообщение; k – текущая длина истории агента a_i .

Локальная память является основным контекстом для принятия решений. Модель, параметризующая политику агента, получает H_i на вход при генерации каждого следующего действия.

Наследование контекста при делегировании составляет второй механизм координации. Родительский агент создает дочернего посредством вызова специализированного инструмента, передавая описание задачи и релевантный фрагмент накопленного контекста. Формально начальная история дочернего агента определяется как

$$H_{child}(0) = \text{delegate}(\text{task}, \text{context}(H_{parent}(t))), \quad (11)$$

где $H_{child}(0)$ – начальная история дочернего агента; task – описание делегируемой задачи, сформулированное родительским агентом; $\text{context}(H_{parent}(t))$ – функция извлечения релевантного контекста из истории родительского агента в момент создания дочернего.

Снимок фиксируется в момент создания и далее не синхронизируется, каждый агент развивает собственную историю независимо. Авторами разработана схема механизма наследования контекста при создании дочернего агента, представленная на рис. 2.

Обмен сообщениями через почтовый ящик составляет третий механизм координации. Каждый агент обладает входящей очередью

$$Q_i = \{(\alpha, \mu, \rho) \mid \alpha \in I, \mu \in W, \rho \in \{0, 1\}\}, \quad (12)$$

где $\alpha \in I$ – агента-отправитель, I – конечное множество агентов, определенное в (2.2); $\mu \in W$ – содержимое сообщения, где W – множество допустимых сообщений; $\rho \in \{0, 1\}$ – признак прочтения сообщения агентом-получателем (0 означает непрочитанное, 1 означает прочитанное).

Отправка сообщения осуществляется операцией

$$\text{send}(\alpha_i, \alpha_j, \mu) \rightarrow Q_j \cup \{(\alpha_i, \mu, 0)\}, \quad (13)$$

где α_i – агент-отправитель; α_j – агент-получатель.

Получение сообщений осуществляется периодическим опросом очереди. Агент вызывает инструмент чтения в рамках своего рабочего цикла, получает релевантные сообщения и самостоятельно решает, какие из них включить в текущую обработку. Такой подход позволяет избежать автоматического накопления всех входящих сообщений в локальной истории и сохранить контекстное окно для приоритетных данных.

Возможны два режима взаимодействия родительского и дочернего агентов. Асинхронное делегирование предполагает, что родитель продолжает собственную работу после создания дочернего агента и обрабатывает его отчет при поступлении. Синхронное ожидание предполагает приостановку работы родителя до получения отчета. Первый режим подходит для фоновой сборки информации, второй необходим, когда результат дочернего агента требуется для принятия следующего решения.

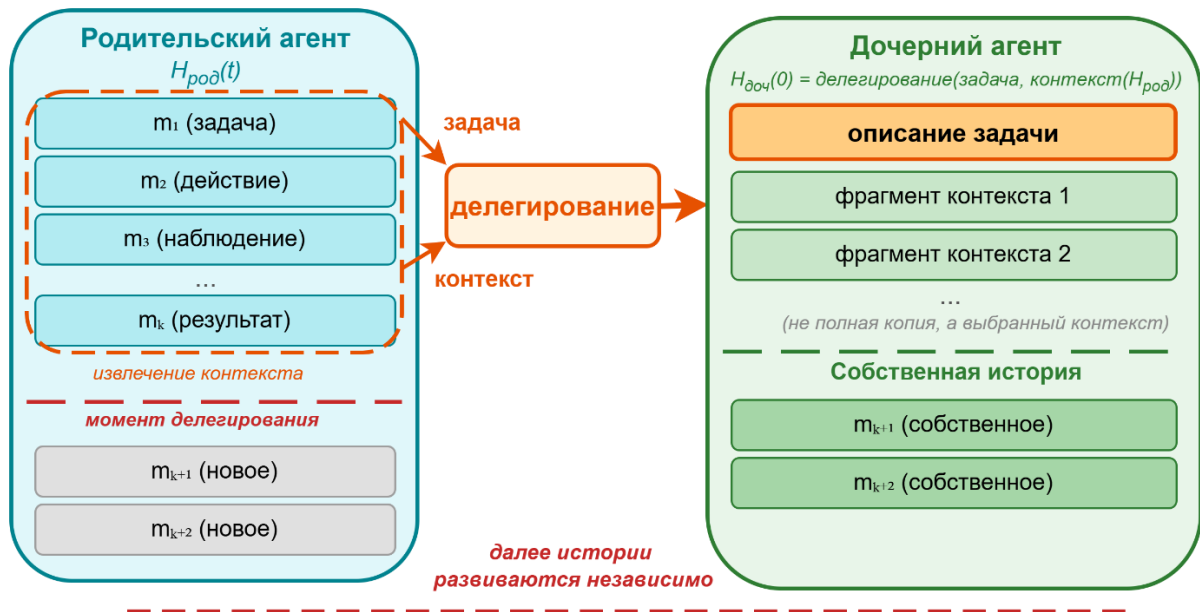


Рис. 2. Наследование контекста при делегировании
Fig. 2. Context inheritance during delegation

Разработано авторами

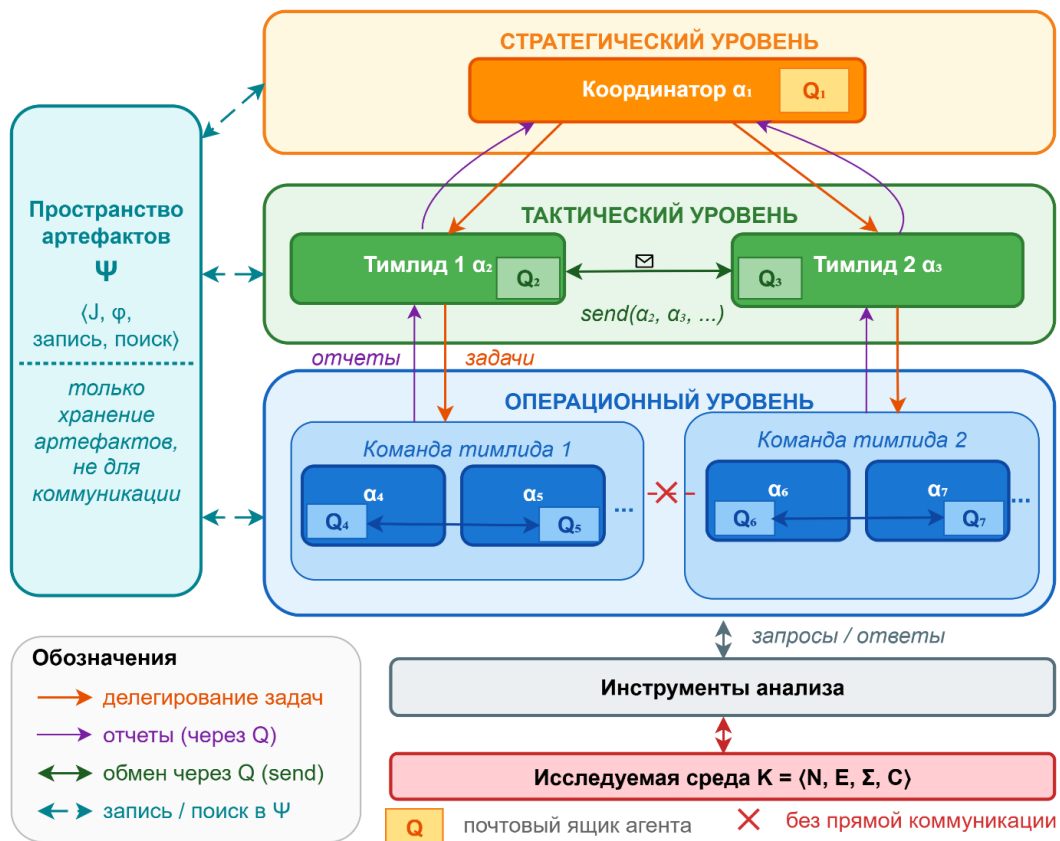


Рис. 3. Механизмы коммуникаций между агентами
Fig. 3. Communication mechanisms between agents

Разработано авторами

Пространство артефактов Ψ , определенное формулой (3), дополняет перечисленные механизмы. Операция записи определяется как

$$write: J \times \Psi \rightarrow \Psi', \quad (14)$$

где Ψ' – обновленное состояние пространства после добавления нового описания ξ и вычисления его векторного представления $\varphi(\xi)$.

Операция поиска определяется как функция

$$search: F \times \Psi \rightarrow 2^J, \quad (15)$$

где F – множество текстовых запросов, а результатом является подмножество описаний из J , ранжированных по семантической близости векторных представлений запроса и записи.

Пространство артефактов не участвует в межагентном обмене сообщениями и служит исключительно для накопления результатов работы и семантического поиска по ним.

Авторами разработана схема коммуникаций между агентами в трехуровневой многоагентной системе (рис. 3).

Совокупность агентов и связей между ними образует направленный граф, который строится эмерджентно в процессе функционирования системы. Автором предлагается формализовать граф делегирования как динамическую структуру, зависящую от текущего шага τ

$$G_{deleg}(\tau) = (N_G(\tau), E_G(\tau), attr(\cdot, \tau)), \quad (16)$$

где $N_G(\tau) \subseteq I$ – множество вершин, каждая из которых соответствует активному агенту системы; $E_G(\tau) \subseteq N_G(\tau) \times N_G(\tau)$ – множество направленных ребер делегирования на шаге τ ; $(\alpha_i, \alpha_j) \in E_G(\tau)$ – ребро, при котором агент α_i создал дочерний агент α_j и делегировал ему подзадачу; $attr(\cdot, \tau)$ – функция разметки вершин, определяемая ниже.

В начальный момент $\tau = 0$ граф содержит единственную вершину α_1 (агент-координатор) и пустое множество ребер: $N_G(0) = \{\alpha_1\}$, $E_G(0) = \emptyset$. По мере делегирования подзадач множества $N_G(\tau)$ и $E_G(\tau)$ расширяются монотонно.

Функция $attr$ сопоставляет каждому агенту его текущую задачу и состояние на шаге τ :

$$attr(\cdot, \tau): N_G(\tau) \rightarrow T \times \mathcal{S}, \quad (17)$$

где T – множество задач, формируемых агентами в ходе анализа (тактические задачи на стратегическом уровне, подзадачи на тактическом); $\mathcal{S} = \{running, waiting, finished, failed\}$ – множество состояний агента в жизненном цикле.

Для каждого агента $\alpha_i \in N_G(\tau)$

$$attr(\alpha_i, \tau) = (t_i(\tau), s_i(\tau)), t_i(\tau) \in T, s_i(\tau) \in \mathcal{S}. \quad (18)$$

Задача $t_i(\tau)$ назначается агенту в момент его создания и остается неизменной на протяжении всего жизненного цикла. Состояние $s_i(\tau)$ обновляется при каждом существенном событии,

таким как начало выполнения задачи, ожидание результатов от дочернего агента, успешное завершение или отказ. Функция $attr$ позволяет определить на любом шаге τ текущее состояние любого агента в графе делегирования, что необходимо для принятия решений вышестоящими агентами, например о создании замены для завершившегося тимлида или о перераспределении задач.

Граф делегирования является деревом с корнем в агенте-координаторе. Многоагентная система является динамической, и ее структура не задается заранее, а формируется в процессе функционирования как следствие решений, принимаемых агентами. Координатор определяет число тактических агентов и задачи для них, исходя из текущей информации об исследуемом комплексе. Тактические агенты аналогичным образом определяют число и задачи операционных агентов. При обнаружении неожиданных компонентов комплекса граф расширяется за счет создания новых агентов. При завершении анализа направления тактический агент завершает работу и его вершина переходит в состояние *finished*. Автором исследована динамика формирования графа делегирования в процессе анализа. Результаты представлены на рис. 4.

Взаимодействие между уровнями архитектуры организовано в виде двух направленных потоков. Нисходящий поток передает задачи от верхних уровней к нижним через механизм делегирования

$$I_{str} \rightarrow I_{tac} \rightarrow I, \quad (18)$$

Восходящий поток передает результаты от нижних уровней к верхним через механизм отчетов о завершении

$$I \rightarrow I_{tac} \rightarrow I_{str}, \quad (19)$$

Нисходящий поток реализуется через инструмент создания дочернего агента. Вышестоящий агент формирует описание задачи, определяет набор инструментов и знаний для дочернего агента и иницирует его создание. Восходящий поток реализуется через отчет о завершении, передаваемый родительскому агенту. Родитель агрегирует результаты от дочерних агентов и включает их в свой контекст.

Каждый агент функционирует в итеративном цикле. На каждом шаге языковая модель генерирует действие a_i на основе политики $\pi_i(H_i)$. Действие может представлять собой вызов одного из доступных инструментов, в том числе чтение очереди сообщений, обращение к пространству артефактов, отправку сообщения другому агенту, создание дочернего агента или вызов внешнего инструмента анализа. Результат выполнения действия возвращается агенту как наблюдение o_i , которое добавляется в локальную историю. Затем

модель генерирует следующее действие, и цикл повторяется до перехода агента в терминальное состояние. Авторами разработана схема основного цикла работы агента, представленная на рис. 5.

Совместная политика системы определяет поведение всей многоагентной системы

$$\pi = (\pi_{str}, \pi_{tac 1}, \dots, \pi_{tac m(t)}, \pi_{oper 1}, \dots, \pi_{oper p(t)}) \quad (20)$$

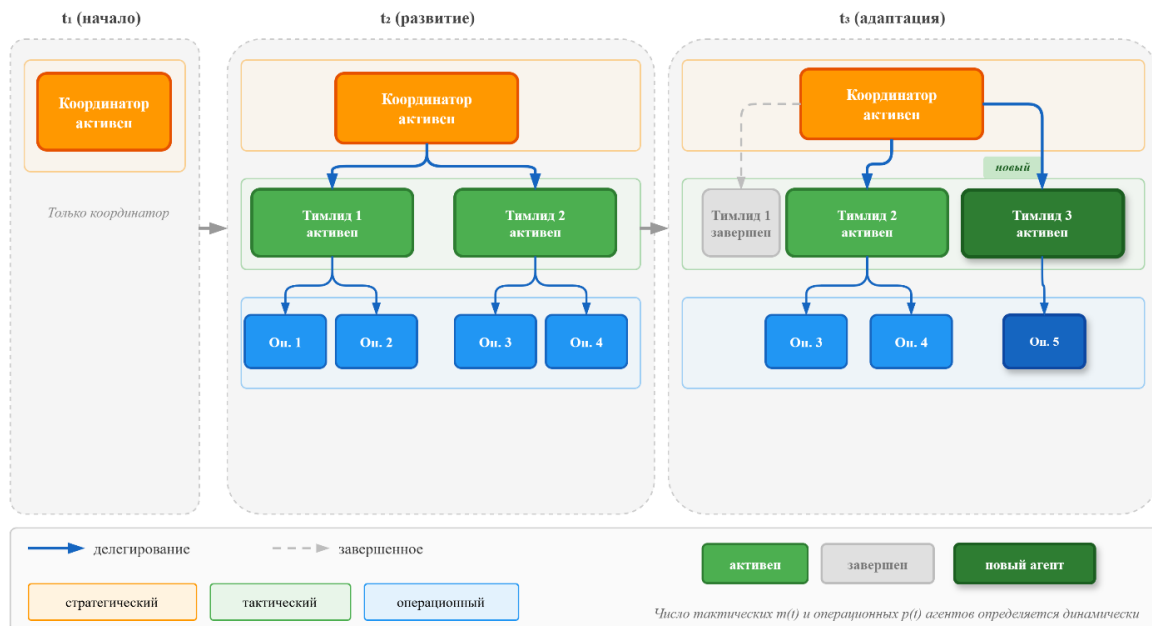


Рис. 4. Динамика формирования графа делегирования
Fig. 4. Dynamics of delegation graph formation

Разработано авторами

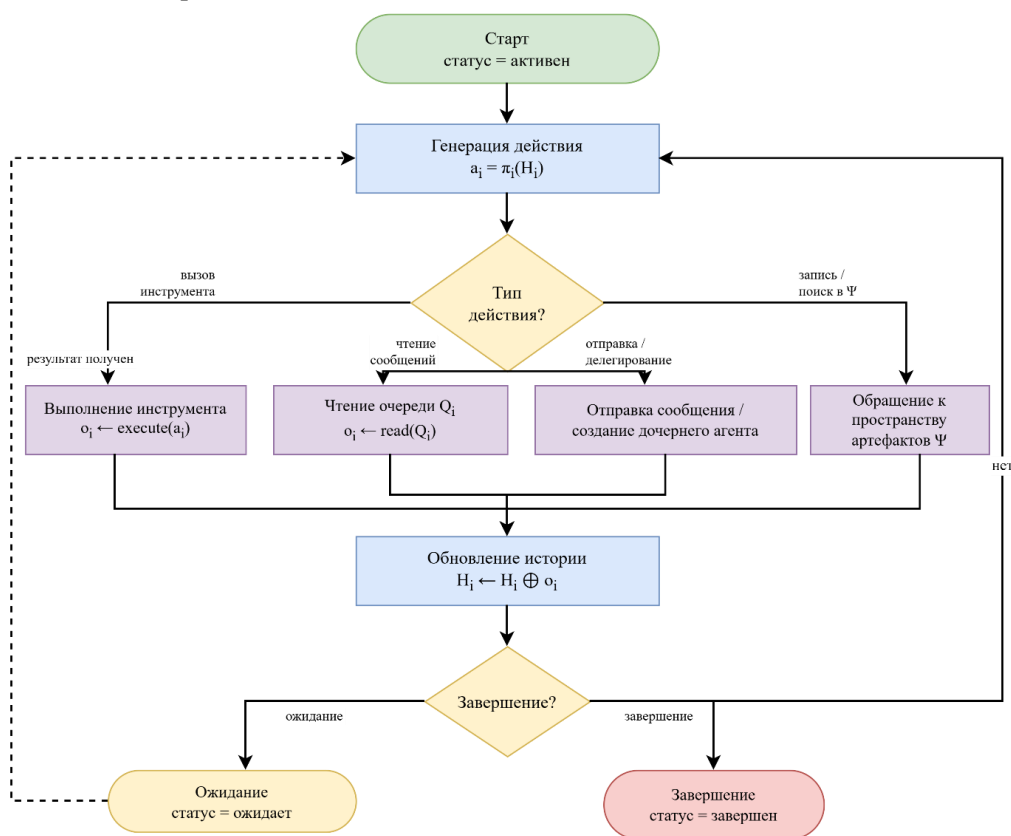


Рис. 5. Основной цикл работы агента
Fig. 5. Main agent work cycle

Разработано авторами

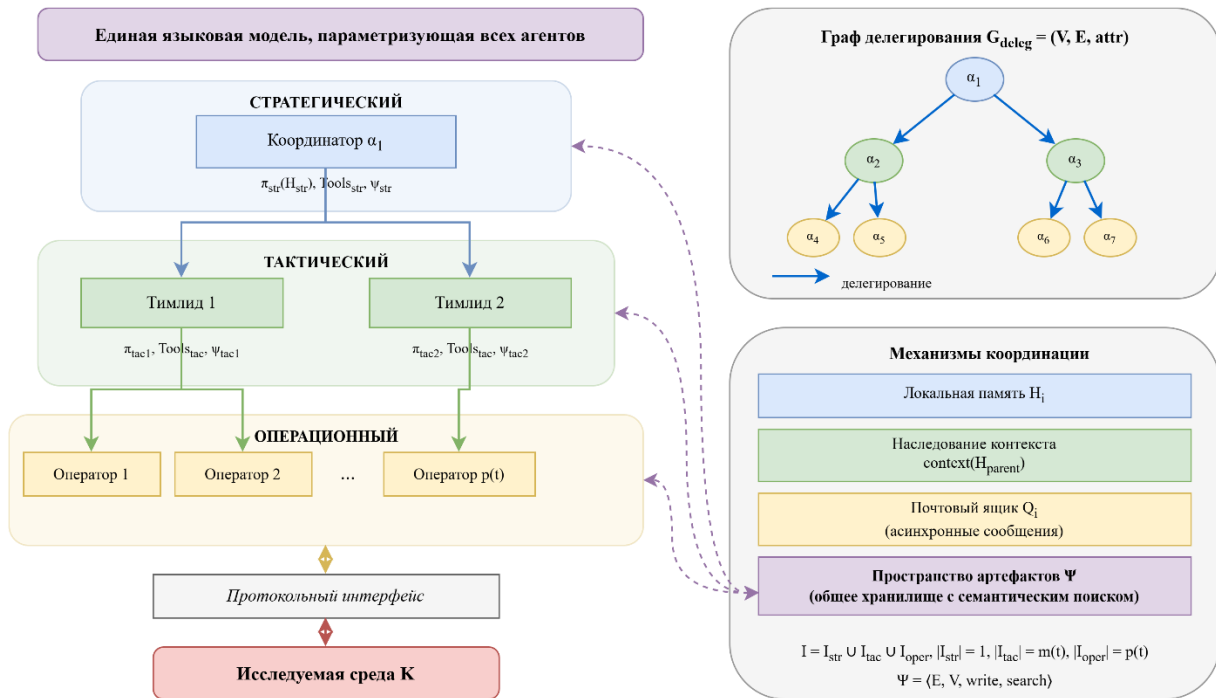


Рис. 6. Обобщающая схема архитектуры многоагентной интеллектуальной системы

Fig. 6. General scheme of the multi-agent intelligent system architecture

Разработано авторами

Число тактических $m(t)$ и операционных агентов $p(t)$ изменяется динамически в процессе работы, поэтому размерность совместной политики не фиксирована.

Авторами разработана обобщающая схема архитектуры многоагентной интеллектуальной системы, представленная на рис. 6. Схема объединяет все введенные ранее компоненты архитектуры и отражает взаимосвязи между иерархией агентов, графом делегирования и механизмами координации, включая пространство артефактов.

ДИСКУССИЯ

Предлагаемая авторская архитектура отличается от существующих подходов к построению многоагентных систем для анализа распределенных сред. Сравнение с альтернативными архитектурами представлено в табл. 2.

Плоская многоагентная система состоит из нескольких равноправных агентов, специализация которых задана при проектировании [4, 7]. Такая система обеспечивает параллелизм, однако не способна к динамической адаптации состава агентов. При увеличении числа участников стоимость координации растет квадратично, поскольку каждый агент должен поддерживать связь с каждым другим. Предлагаемая архитектура устраняет эти ограничения за счет трехуровневой декомпозиции, динамического формирования графа делегирования и сочетания нескольких механизмов координации.

Сравнение подходов к организации информационного пространства также свидетельствует в пользу предлагаемой архитектуры (табл. 3). Централизованное хранилище по паттерну общей доски обеспечивает высокую согласованность, однако создает единую точку отказа и плохо масштабируется. Чистый обмен сообщениями без общего хранилища масштабируется хорошо, но не гарантирует согласованность и затрудняет горизонтальный обмен результатами. Предлагаемая гибридная модель сочетает достоинства обоих подходов, используя локальную память и адресные сообщения для оперативной координации, а пространство артефактов для накопления и горизонтального обмена результатами.

Вместе с тем необходимо отметить ограничения предлагаемого подхода. Качество работы системы существенно зависит от возможностей языковой модели, параметризующей политики агентов. Модель должна эффективно работать на различных уровнях абстракции, от стратегического планирования до интерпретации результатов атомарных операций. Настоящая работа определяет архитектуру системы, но не содержит экспериментальной валидации на реальных распределенных комплексах. Экспериментальное исследование предполагается в дальнейших публикациях. Кроме того, границы применимости трех

Таблица 2

Сравнение архитектур многоагентных систем
Table 2. Comparison of multi-agent system architectures

Характеристика	Плоская многоагентная система	Предлагаемая иерархическая архитектура
Число уровней	1	3, частный случай n-уровневой архитектуры
Специализация агентов	Фиксированная, задана при проектировании	Динамическая, определяется при создании
Информационное пространство	Общая память или прямой обмен	Гибридное, четыре механизма
Координация	Прямой обмен между равноправными агентами	Иерархическая делегация, почтовый ящик и пространство артефактов
Масштабируемость	Средняя, ограничена стоимостью координации	Высокая, каждый уровень масштабируется независимо
Адаптивность	Ограничена фиксированным составом	Высокая, эмерджентное формирование состава

Разработано авторами

Таблица 3

Сравнение подходов к организации информационного пространства
Table 3. Comparison of information space organization approaches

Подход	Масштабируемость	Согласованность	Устойчивость
Централизованное хранилище (паттерн общей доски)	Низкая	Высокая	Низкая
Прямой обмен сообщениями без общего хранилища	Высокая	Не гарантируется	Высокая
Предлагаемая архитектура	Высокая	Обеспечивается сочетанием локальных и централизованного механизмов	Высокая

Разработано авторами

уровневой конфигурации определяются структурой задачи, и для задач с иной глубиной декомпозиции оптимальное значение n может отличаться от трех.

ЗАКЛЮЧЕНИЕ

Проведенное исследование позволило решить задачу формализации иерархической организации многоагентной интеллектуальной системы, ориентированной на автоматизированный анализ распределенных программно-аппаратных комплексов. Теоретическое обоснование подтвердило оптимальность трехуровневой конфигурации ($n = 3$), при которой стратегический уровень обеспечивает декомпозицию задачи, тактический уровень координирует направления анализа, а операционный уровень выполняет непосредственное взаимодействие с исследуемой средой.

Параметрическая модель агента, представленная как кортеж системной инструкции и набора доступных инструментов при единой языковой модели, обеспечивает функциональную специализацию без необходимости обучения отдельных

моделей для каждой роли. Сравнительный анализ с плоскими, цепочечными и полносвязными архитектурами показал, что предложенная гибридная схема координации сочетает управляемость централизованных схем с гибкостью горизонтального взаимодействия на тактическом уровне.

Граф делегирования, формируемый решениями агентов в процессе функционирования, выступает эмерджентной структурой, отражающей адаптацию системы к конкретному комплексу. Пространство артефактов дополняет локальные механизмы координации глобальным семантическим поиском по накопленным результатам. Дальнейшее исследование предполагает разработку системы критериев оценки качества поведенческих стратегий агентов и метода кросс-уровневого обучения с экспериментальной валидацией на распределенных программно-аппаратных комплексах.

Авторы заявляют об отсутствии конфликта интересов, требующего раскрытия в данной статье.

The authors declare the absence a conflict of interest warranting disclosure in this article.

ЛИТЕРАТУРА

1. Астраханцев Р.Г., Бобков С.П. Методологические основы построения многоагентной интеллектуальной системы. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2026. № 1(67). С. 127–136.
2. Нигматуллин Р.Р. Системный анализ как направление научных исследований. *Вестник Российской академии наук*. 2007. Т. 77, № 2. С. 101–107.
3. Dorri A., Kanhere S.S., Jurdak R. Multi-Agent Systems: A Survey. *IEEE Access*. 2018. Vol. 6. P. 28573–28593.
4. Принев М.А., Леденева Т.М., Гаршина В.В. Многоагентные системы: обзор современных подходов к моделированию и проектированию (Часть 2). *Вестник ВГУ. Серия: Системный анализ и информационные технологии*. 2024. № 4. С. 167–190.
5. Jennings N.R., Wooldridge M. Agent-Oriented Software Engineering. *Artificial Intelligence*. 2000. Vol. 117. P. 277–296.
6. D’Aniello G., De Falco M., Mastrandrea N. Designing a multi-agent system architecture for managing distributed operations within cloud manufacturing. *Evolutionary Intelligence*, 2020.
7. Астраханцева И.А., Астраханцев Р.Г., Бобков С.П. Агентные интеллектуальные системы с точки зрения системного анализа. *Современные наукоемкие технологии. Региональное приложение*. 2025. № 4(84). С. 101–109.
8. Астраханцева И.А., Горев С.В., Астраханцев Р.Г. Системный подход к анализу фрактальной природы сложных технических систем. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2023. № 3(57). С. 89–97.
9. Астраханцева И.А., Горев С.В., Астраханцев Р.Г. Фрактальный анализ в оценке эффективности и надежности сложных технических систем. *Современные наукоемкие технологии. Региональное приложение*. 2023. № 4(76). С. 60–68.
10. Месарович М., Мако Д., Такахара И. Теория иерархических многоуровневых систем. М.: Мир, 1973. 344 с.
11. Бурков В.Н., Новиков Д.А. Теория активных систем: состояние и перспективы. М.: Синтег, 1999. 128 с.
12. Bernstein D.S., Givan R., Immerman N., Zilberstein S. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research*. 2002. Vol. 27, N 4. P. 819–840.
13. Oliehoek F.A., Spaan M.T.J., Vlassis N. A Concise Introduction to Decentralized POMDPs. *Proceedings of AAMAS 2013*. 2013. P. 1257–1264.
14. Busoniu L., Babuska R., De Schutter B. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 2008, Vol. 38, N 2. P. 156–172.
15. Koops W., Junges S., Jansen N. Approximate Dec-POMDP Solving Using Multi-Agent A*. *Proceedings of ICAPS 2024*. 2024. P. 745–753.
16. Astrakhantsev R., Chuhno A., Dmukh A. et al. Differences with high probability and impossible differentials for the KB-256 cipher. *Journal of Computer Virology and Hacking Techniques*. 2024. Vol. 20, N 3. P. 525–531.
17. ГОСТ Р 59277–2020. Информационные технологии. Многоагентные системы. Термины и определения. Введ. 2020-09-01. М.: Стандартинформ, 2020. 12 с.
18. Bobkov S.P., Astrakhantseva I.A. The use of multi-agent systems for modeling technological processes. *Journal of Physics: Conference Series*. 2021. Vol. 2001. P. 012002.

REFERENCES

1. Astrakhantsev R.G., Bobkov S.P. Foundations of multi-agent intelligent system design. *Ivecofin*. 2026. N 1(67). P. 127–136.
2. Nigmatullin R.R. Systems analysis as a research direction. *Vestnik Rossiiskoi Akademii Nauk*. 2007. Vol. 77, N 2. P. 101–107.
3. Dorri A., Kanhere S.S., Jurdak R. Multi-Agent Systems: A Survey. *IEEE Access*. 2018. Vol. 6. P. 28573–28593.
4. Prinev M.A., Ledeneva T.M., Garshina V.V. Multi-agent systems: a review of modern approaches to modeling and design (Part 2). *Vestnik VGU. Systems Analysis and Information Technologies*. 2024. N 4. P. 167–190.
5. Jennings N.R., Wooldridge M. Agent-Oriented Software Engineering. *Artificial Intelligence*. 2000. Vol. 117. P. 277–296.
6. D’Aniello G., De Falco M., Mastrandrea N. Designing a multi-agent system architecture for managing distributed operations within cloud manufacturing. *Evolutionary Intelligence*, 2020.
7. Astrakhantseva I.A., Astrakhantsev R.G., Bobkov S.P. Agent intelligent systems from the perspective of systems analysis. *Modern high technology. Regional application*. 2025, N 4(84). P. 101–109.
8. Astrakhantseva I.A., Gorev S.V., Astrakhantsev R.G. A systemic approach to the analysis of the fractal nature of complex technical systems. *Ivecofin*. 2023. N 3(57). P. 89–97.
9. Astrakhantseva I.A., Gorev S.V., Astrakhantsev R.G. Fractal analysis in assessing the efficiency and reliability of complex technical systems. *Modern high technology. Regional application*. 2023. N 4(76). P. 60–68.
10. Mesarovic M., Macko D., Takahara Y. Theory of hierarchical, multilevel, systems. Moscow, Mir, 1973. 344 p.
11. Burkov V.N., Novikov D.A. Theory of active systems: state and prospects. Moscow, Sinteg, 1999. 128 p.
12. Bernstein D.S., Givan R., Immerman N., Zilberstein S. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research*. 2002. Vol. 27, N 4. P. 819–840.
13. Oliehoek F.A., Spaan M.T.J., Vlassis N. A Concise Introduction to Decentralized POMDPs. *Proceedings of AAMAS 2013*. 2013. P. 1257–1264.
14. Busoniu L., Babuska R., De Schutter B. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 2008, Vol. 38, N 2. P. 156–172.
15. Koops W., Junges S., Jansen N. Approximate Dec-POMDP Solving Using Multi-Agent A*. *Proceedings of ICAPS 2024*. 2024. P. 745–753.
16. Astrakhantsev R., Chuhno A., Dmukh A. et al. Differences with high probability and impossible differentials for the KB-256 cipher. *Journal of Computer Virology and Hacking Techniques*. 2024. Vol. 20, N 3. P. 525–531.
17. GOST R 59277–2020. Information Technologies. Multi-Agent Systems. Terms and Definitions. Moscow, Standartinform, 2020. 12 p.
18. Bobkov S.P., Astrakhantseva I.A. The use of multi-agent systems for modeling technological processes. *Journal of Physics: Conference Series*. 2021. Vol. 2001. P. 012002.

Поступила в редакцию (Received) 21.03.2026

Принята к опубликованию (Accepted) 18.05.2026