

**МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ
МНОГОАГЕНТНОЙ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ****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, Шереметевский пр., 7, Иваново, 153000, Россия

E-mail: rgastrakhtantsev@gmail.com*, bsp@isuct.ru

Статья посвящена разработке методологических основ построения многоагентной интеллектуальной системы для автоматизированного анализа состояния распределенных программно-аппаратных комплексов в условиях неопределенности и частичной наблюдаемости среды. Дано формальное определение распределенного программно-аппаратного комплекса как кортежа, включающего множества вычислительных узлов, сетевых соединений, программных служб и конфигурационных параметров. Разработана формальная модель задачи автоматизированного анализа на основе расширения математического аппарата децентрализованных марковских процессов принятия решений в условиях частичной наблюдаемости. В отличие от классической постановки, авторская модель включает композитное пространство состояний, учитывающее как параметры исследуемой среды, так и накопленные агентами сведения, динамическое пространство действий, расширяющееся по мере получения новой информации, а также механизм верификации достижения целевого состояния. Разработана трехуровневая иерархическая архитектура многоагентной интеллектуальной системы, включающая стратегический уровень (агент-координатор), тактический уровень (агенты-тимлиды) и операционный уровень (микро-агенты). В отличие от статичных архитектур с фиксированным составом, предложенная архитектура обеспечивает динамическое формирование агентов в процессе функционирования. Описаны механизмы координации через иерархическую передачу задач и общее информационное пространство. Предложенные методологические основы являются универсальными и применимы к широкому классу задач автоматизированного исследования распределенных программно-аппаратных сред.

Ключевые слова: многоагентная интеллектуальная система, распределенный программно-аппаратный комплекс, системный анализ, иерархическая декомпозиция, частичная наблюдаемость, марковский процесс принятия решений, автоматизированный анализ состояния.

FOUNDATIONS OF MULTI-AGENT INTELLIGENT SYSTEM DESIGN**R.G. Astrakhtantsev, S.P. Bobkov**

Roman G. Astrakhtantsev* (ORCID 0000-0001-9880-2826), Sergey P. Bobkov (ORCID 0000-0001-7315-1625)

Ivanovo State University of Chemistry and Technology, Sheremetevsky avenue, 7, Ivanovo, 153000, Russia

E-mail: rgastrakhtantsev@gmail.com*, bsp@isuct.ru

This paper develops methodological foundations for constructing a multi-agent intelligent system designed for automated state analysis of distributed software and hardware complexes operating under uncertainty and partial observability. A distributed software and hardware complex is formally defined as a tuple of computational nodes, network connections, software services, and configuration parameters. The automated analysis problem is then formalized by extending the decentralized partially observable Markov decision process framework in three respects. First, a composite state space is introduced that

captures both the target environment and the information resources accumulated by the agents. Second, a dynamic action space is defined that expands as agents acquire new knowledge about the complex. Third, a reward verification mechanism is incorporated to ensure reliable feedback under sparse reward conditions. Building on this formal model, a three-level hierarchical architecture is proposed in which a single strategic coordinator delegates tasks to dynamically instantiated tactical team leaders, who in turn manage operational micro-agents responsible for direct interaction with analytical tools. Unlike flat or fixed-composition multi-agent designs, the architecture supports on-the-fly agent creation and termination, enabling the system to adapt its structure to the current stage of analysis. Inter-level coordination is achieved through hierarchical task assignment combined with a shared information space accessible to all agents. The resulting methodology is domain-independent and applicable to a broad class of problems involving automated exploration of distributed software-hardware environments.

Keywords: multi-agent intelligent system, distributed software and hardware complex, systems analysis, hierarchical decomposition, partial observability, Markov decision process, automated state analysis.

Для цитирования:

Астраханцев Р.Г., Бобков С.П. Методологические основы построения многоагентной интеллектуальной системы. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2026. № 01(67). С. 127-136. DOI: 10.6060/ivecofin.2026671.767

For citation:

Astrakhtantsev R.G., Bobkov S.P. Foundations of multi-agent intelligent system design. *Ivecofin*. 2026. N 01(67). P. 127-136. DOI: 10.6060/ivecofin.2026671.767 (in Russian)

ВВЕДЕНИЕ

Задачи автоматизированного анализа состояния распределенных программно-аппаратных комплексов занимают одно из центральных мест в системном анализе [1, 2]. Современные распределенные комплексы объединяют десятки и сотни вычислительных узлов с разнородным программным обеспечением, а их конфигурация изменяется в процессе функционирования [3, 4]. Определение текущего состояния такого комплекса представляет собой задачу последовательного принятия решений, при которой информация о среде раскрывается постепенно и является принципиально неполной.

Под распределенным программно-аппаратным комплексом авторами предлагается понимать сложную техническую систему, состоящую из территориально рассредоточенных вычислительных узлов, объединенных сетевой инфраструктурой и функционирующих как единое целое для решения прикладных задач. Формально распределенный программно-аппаратный комплекс определяется как кортеж

$$K = \langle N, E, S, C \rangle, \quad (1)$$

где N – конечное множество вычислительных узлов;

$E \subseteq N \times N$ – множество сетевых соединений между узлами;

S – множество программных служб, функционирующих на узлах;

C – множество конфигурационных параметров, определяющих состояние комплекса в момент времени t .

Примерами таких комплексов служат корпоративные вычислительные сети, промышленные системы управления, территориально распределенные инфраструктуры обработки данных [3, 5]. Задача анализа состояния подобного комплекса состоит в определении текущей конфигурации узлов, выявлении активных служб, идентификации отклонений параметров от нормативных значений и верификации полученных результатов.

Существующие подходы к автоматизированному анализу распределенных сред опираются преимущественно на одноагентные системы или плоские многоагентные архитектуры с фиксированным составом [6, 7, 8]. Одноагентные системы ограничены в масштабируемости и не позволяют эффективно декомпозировать сложные задачи исследования. Плоские многоагентные архитектуры затрудняют координацию при увеличении числа агентов вследствие необходимости прямого взаимодействия каждого агента со всеми остальными [9]. Подходы на основе иерархической декомпозиции с динамическим формированием состава агентов представлены в научной литературе недостаточно, особенно в части строгого математического обоснования структуры и механизмов координации [8, 10].

1. ФОРМАЛЬНАЯ МОДЕЛЬ ЗАДАЧИ АВТОМАТИЗИРОВАННОГО АНАЛИЗА СОСТОЯНИЯ РАСПРЕДЕЛЕННЫХ ПРОГРАММНО-АППАРАТНЫХ КОМПЛЕКСОВ

Автоматизированный анализ состояния распределенного программно-аппаратного комплекса по своей природе является задачей последовательного принятия решений в условиях значительной неопределенности. Агентская система обладает лишь частичной информацией о состоянии исследуемого комплекса. Каждое действие предоставляет новые, зачастую неполные сведения о среде.

Простейшей моделью для задач последовательного принятия решений является марковский процесс принятия решений. Он предполагает, что агент обладает полной и достоверной информацией о состоянии среды. Однако для рассматриваемой задачи - это допущение неприемлемо [11, 12]. Информация об исследуемом комплексе раскрывается постепенно в результате активных действий агентов. Частично наблюдаемый марковский процесс принятия решений моделирует ситуацию, когда агент получает лишь косвенные наблюдения, позволяющие строить вероятностные предположения об истинном состоянии среды [13]. Вместе с тем разрабатываемая система является многоагентной, и множество специализированных агентов действует в общей среде, обладая только локальной информацией [14, 15]. Это требует перехода к децентрализованной модели. Наиболее адекватным математическим аппаратом для формализации рассматриваемой задачи является децентрализованный частично наблюдаемый марковский процесс принятия решений [13, 16].

Указанный процесс определяется кортежем из девяти элементов

$$M = \langle I, S, \{A_i\}_{i \in I}, T, R, \{\Omega_i\}_{i \in I}, O, \gamma, b_0 \rangle, \quad (2)$$

где I — конечное множество агентов;

$|I| = n$, S — конечное множество состояний среды;

$\{A_i\}_{i \in I}$ — множества действий для каждого агента i , совместное действие $a = (a_1, \dots, a_n) \in A = A_1 \times \dots \times A_n$;

T — функция переходов, определяющая вероятность $T(s' | s, a)$ перехода из состояния s в состояние s' при совместном действии a ;

R — функция вознаграждения;

$\{\Omega_i\}_{i \in I}$ — множества наблюдений для каждого агента;

O — функция наблюдений, определяющая вероятность $O(o | s', a)$ получения наблюдения o после перехода в состояние s' ;

γ — коэффициент дисконтирования, $\gamma \in [0, 1]$;

b_0 — начальное распределение вероятностей по состояниям.

Поскольку истинное состояние s недоступно агентам напрямую, система поддерживает состояние веры $b(s)$, представляющее собой распределение вероятностей по множеству возможных состояний. Обновление состояния веры после выполнения действия a и получения наблюдения o осуществляется по правилу Байеса с учетом функций переходов и наблюдений [11, 13]. Политика агента i определяется как отображение истории наблюдений и действий в распределение вероятностей над множеством действий

$$\pi_i : (\Omega_i \times A_i)^* \cdot \Omega_i \rightarrow \Delta(A_i), \quad (3)$$

где $(\Omega_i \times A_i)^*$ — множество конечных последовательностей пар наблюдений и действий агента i ;

$\Delta(A_i)$ — множество распределений вероятностей над действиями. Совместная политика $\pi = (\pi_1, \dots, \pi_n)$ определяет поведение всей многоагентной системы.

Целью является нахождение оптимальной совместной политики, максимизирующей математическое ожидание суммарного дисконтированного вознаграждения

$$\pi^* = \arg \max_{\pi} E_{\pi} [\sum_{t=0}^{\infty} \gamma^t \cdot R(s_t, a_t) | b_0]. \quad (4)$$

Классическая модель не учитывает ряд существенных особенностей задачи анализа распределенных программно-аппаратных комплексов. В стандартной постановке состояние описывает исключительно среду, пространство действий является статичным, а вознаграждение наблюдается непосредственно. В реальной задаче автоматизированного анализа эти допущения не выполняются [5, 17, 18]. Состояние системы включает не только характеристики исследуемого комплекса, но и накопленные агентами информационные ресурсы. Множество доступных действий расширяется по мере получения новых сведений о среде.

Авторами предлагается расширение классической модели, устраняющее указанные ограничения. Авторская модель отличается от классической в трех аспектах, а именно в структуре пространства состояний, в формировании множества допустимых действий и в механизме вычисления вознаграждения.

Первое расширение касается пространства состояний. В авторской модели состояние является композитным и включает две компоненты

$$S = S_{env} \cdot S_{agent}, \quad (5)$$

где S_{env} — состояние исследуемого программно-аппаратного комплекса K , включающее параметры

вычислительных узлов N , активные сетевые соединения E и конфигурацию программных служб S ;

Sagent – внутреннее состояние агентской системы, включающее накопленные сведения о среде, полученные параметры доступа к компонентам комплекса и собранные информационные артефакты.

Такое представление позволяет явно моделировать тот факт, что действия агентов изменяют не только исследуемый комплекс, но и информационные ресурсы самой агентской системы. Подобная двойственность является принципиальным свойством задачи автоматизированного анализа, в которой получение сведений об одном компоненте среды создает предпосылки для исследования других компонентов [3, 4].

Второе расширение определяет динамический характер множества доступных действий

$$A(t) = f(S_{agent}(t)), \quad (6)$$

где $A(t)$ – множество доступных действий в момент времени t ;

f – функция, определяющая зависимость пространства действий от текущего состояния агентской системы. При этом в типичном случае выполняется свойство монотонного расширения

$$A(t) \subseteq A(t + 1). \quad (7)$$

Получение новых сведений и расширенного доступа к компонентам комплекса приводит к расширению множества допустимых действий. Данное свойство отражает логику последовательного исследования, когда анализ одного компонента открывает доступ к ранее недоступ-

ным компонентам [5, 18]. Вместе с тем при наличии активных защитных или компенсационных механизмов на стороне исследуемого комплекса возможно и сужение множества действий вследствие обнаружения и блокирования определенных воздействий.

Третье расширение касается функции вознаграждения. В задаче автоматизированного анализа вознаграждение характеризуется разреженностью, и основное вознаграждение выдается только при достижении ключевых целей исследования. Для обеспечения корректности обучения вводится система верификации V , подтверждающая факт достижения целевого состояния

$$R(s, a) = R_{base}(s, a) + R_{goal} \cdot 1[V(s, a) = True], \quad (8)$$

где $R_{base}(s, a)$ – базовое вознаграждение за выполнение действия, которое может учитывать промежуточный прогресс;

R_{goal} – вознаграждение за достижение цели исследования, существенно превышающее базовое;

$1[\cdot]$ – индикаторная функция, принимающая значение единица при выполнении условия;

$V(s, a)$ – функция верификации, подтверждающая достижение целевого состояния.

Система верификации может быть реализована как автоматическая проверка, например, путем сопоставления полученных параметров с эталонными значениями, или как подтверждение со стороны эксперта-оператора [19, 20].

Отличия авторской модели от классической постановки систематизированы в табл. 1.

Таблица 1. Сравнение классической и авторской моделей децентрализованного частично наблюдаемого марковского процесса принятия решений

Table 1. Comparison of classical and proposed models of decentralized partially observable Markov decision process

Параметр модели	Классическая модель	Авторская модель
Пространство состояний	Описывает только среду	Композитное, состоит из состояния среды и состояния агентской системы (ф-ла 5)
Пространство действий	Статичное, задано заранее	Динамическое, зависит от накопленных сведений (ф-ла 6)
Функция вознаграждения	Наблюдается непосредственно	Требует внешней верификации достижения цели (ф-ла 8)

Источник: разработано авторами
Source: developed by the authors

Пространство состояний задачи анализа распределенного программно-аппаратного комплекса растет экспоненциально с увеличением числа вычислительных узлов

$$|S| = O(2^{k \cdot N}), \quad (9)$$

где N – число узлов в исследуемом комплексе;

k – число бинарных признаков, описывающих состояние каждого узла (такие как «обнаружен», «параметры определены», «доступ получен», «конфигурация верифицирована»). При $k = 4$ признаках и $N = 10$ узлах число возможных состояний достигает 2^{40} , что превышает 10^{12} . Данная вычислительная сложность обосновывает необходимость применения приближенных методов решения, в частности, методов обучения с подкреплением [14, 16].

Таким образом, предлагаемая авторская модель расширяет классический аппарат децентрализованного частично наблюдаемого марковского процесса принятия решений в трех направлениях. Введение композитного состояния позволяет учитывать накопление информационных ресурсов агентской системой в ходе исследования. Динамическое пространство действий отражает расширение возможностей агентов по мере получения новых сведений об исследуемом комплексе. Механизм верификации обеспечивает корректность обратной связи при обучении в условиях разреженного вознаграждения. Модель определяет целевой функционал оптимизации и формализует особенности предметной области. Для практической реализации необходимо определить структуру многоагентной системы и механизмы взаимодействия агентов.

2. АРХИТЕКТУРА МНОГОАГЕНТНОЙ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ

Для практической реализации авторской оптимизационной модели необходимо разработать архитектуру многоагентной системы, определяющую структуру множества агентов и механизмы их координации.

Авторами предлагается понимать под многоагентной интеллектуальной системой организованную в виде иерархической или сетевой структуры совокупность автономных программных сущностей (агентов), обладающих способностью к восприятию и интерпретации 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 [6], где многоагентная система трактуется как «система, состоящая из множества взаимодействующих интеллектуальных агентов». Однако авторское определение является более широким и акцентирует внимание на двух принципиальных аспектах. Во-первых, на интеллектуальном характере агентов, то есть способности к автономному решению задач и системному эффекту от их объединения. Во-вторых, на функционировании в стохастических условиях с частичной неопределенностью, что позволяет более точно описывать системы, применяемые в реальных, динамически изменяющихся средах [7, 9].

В рамках системного анализа такая система рассматривается как динамически изменяющаяся кибернетическая система с распределенным управлением, множеством контуров обратной связи, критериальной многомерностью и возможностью синтеза сложных поведенческих стратегий через самоорганизацию и согласование между агентами [1, 2, 17].

Проектирование архитектуры многоагентной системы для задачи анализа распределенных программно-аппаратных комплексов опирается на принципы системного анализа [1, 2]. Задача анализа является сложной и характеризуется многообразием требуемых компетенций, различными временными масштабами принятия решений и необходимостью координации множества параллельных процессов. Принцип иерархической декомпозиции позволяет разбить сложную систему на уровни с различной степенью абстракции [3]. Принцип модульности обеспечивает независимость компонентов и возможность их замены [4]. Принцип единого информационного пространства создает основу для координации без жесткой связности [15, 20–21].

Авторами предлагается разбить множество агентов системы на три непересекающихся подмножества в соответствии с уровнем иерархии

$$I = I_{str} \cup I_{tac} \cup I_{oper}, \quad (10)$$

где I_{str} – множество агентов стратегического уровня;

I_{tac} – множество агентов тактического уровня;

I_{oper} – множество агентов операционного уровня. Пересечение любых двух подмножеств пусто, что обеспечивает однозначность принадлежности каждого агента определенному уровню. Мощности указанных множеств определяются следующим образом

$$|I_{str}| = 1, |I_{tac}| = m(t), |I_{oper}| = n(t), \quad (11)$$

где стратегический уровень содержит ровно одного агента-координатора;

$m(t)$ – число агентов тактического уровня в момент времени t ;

$n(t)$ – число агентов операционного уровня в момент времени t . Тактический и операционный уровни формируются динамически в зависимости от текущей задачи и хода исследования.

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

$$f_{str} : G \cdot H_{str} \rightarrow T, \quad (12)$$

где G – множество глобальных целей исследования;

H_{str} – история взаимодействий стратегического агента, включающая реестр созданных агентов тактического уровня, назначенные им задачи и полученные отчеты;

T – множество тактических задач. Агент стратегического уровня анализирует поставленную цель, оценивает накопленную информацию об исследуемом комплексе и формирует набор тактических задач для агентов нижележащего уровня.

Тактический уровень состоит из агентов-тимлидов, каждый из которых специализируется на определенном направлении исследования. Функция тактического уровня определяется как отображение

$$f_{tac} : T \cdot \Psi \cdot H_{tac} \rightarrow A^*, \quad (13)$$

где T – тактическая задача, полученная от стратегического уровня;

Ψ – общее информационное пространство;

H_{tac} – история взаимодействий тактического агента с операционным уровнем и накопленные результаты;

A^* – последовательность действий или подзадач для операционного уровня. Специализация каждого тимлида определяется координатором в момент создания на основе анализа цели и накопленной информации. Состав и число тимлидов изменяются в ходе выполнения задачи.

Операционный уровень состоит из микро-агентов, выполняющих атомарные операции. Функция операционного уровня определяется как отображение

$$f_{oper} : A \cdot H_{oper} \rightarrow O, \quad (14)$$

где A – конкретное действие, назначенное тактическим уровнем;

H_{oper} – история взаимодействий микро-агента с инструментами и средой, включающая предыдущие попытки, полученные ошибки и промежуточные результаты;

O – наблюдение, полученное в результате выполнения действия. Микро-агенты непосредственно взаимодействуют с инструментами исследования и возвращают результаты вышестоящим уровням.

Авторская трехуровневая архитектура многоагентной системы представлена на рис. 1.

Общее информационное пространство служит централизованным хранилищем данных об исследуемом комплексе и артефактов, накопленных в ходе анализа. Формально оно определяется как кортеж

$$\Psi = (K, V), \quad (15)$$

где K – множество ключей, идентифицирующих информационные объекты,

V – множество значений, хранящих данные произвольного типа. Информационное пространство содержит результаты исследования структуры среды, идентифицированные параметры состояния компонентов, полученные параметры доступа и иные артефакты, накопленные агентами в процессе анализа. Любой агент системы может записывать свои находки в хранилище и читать данные, необходимые для выполнения текущей задачи.

Координация в системе осуществляется посредством двух механизмов. Иерархическое взаимодействие обеспечивает передачу задач от верхних уровней к нижним и возврат результатов в обратном направлении. Нисходящий поток передает задачи по маршруту от стратегического уровня через тактический к операционному. Восходящий поток передает результаты выполнения от операционного уровня через тактический к стратегическому. Внутри команды одного тимлида микро-агенты взаимодействуют напрямую, обмениваясь сообщениями для согласования действий и передачи промежуточных результатов.

Схема потоков данных между уровнями и взаимодействия с общим информационным пространством представлена на рис. 2.

Связь предлагаемой архитектуры с авторской оптимизационной моделью устанавливается через понятие политики агента. Политика каждого агента определяется функцией соответствующего уровня

$$\pi_i = f_{level}(i), \quad (16)$$

где π_i – политика агента i согласно определению из формулы (3),

$f_{level}(i)$ – функция уровня, которому принадлежит агент, согласно формулам (12), (13) или (14).

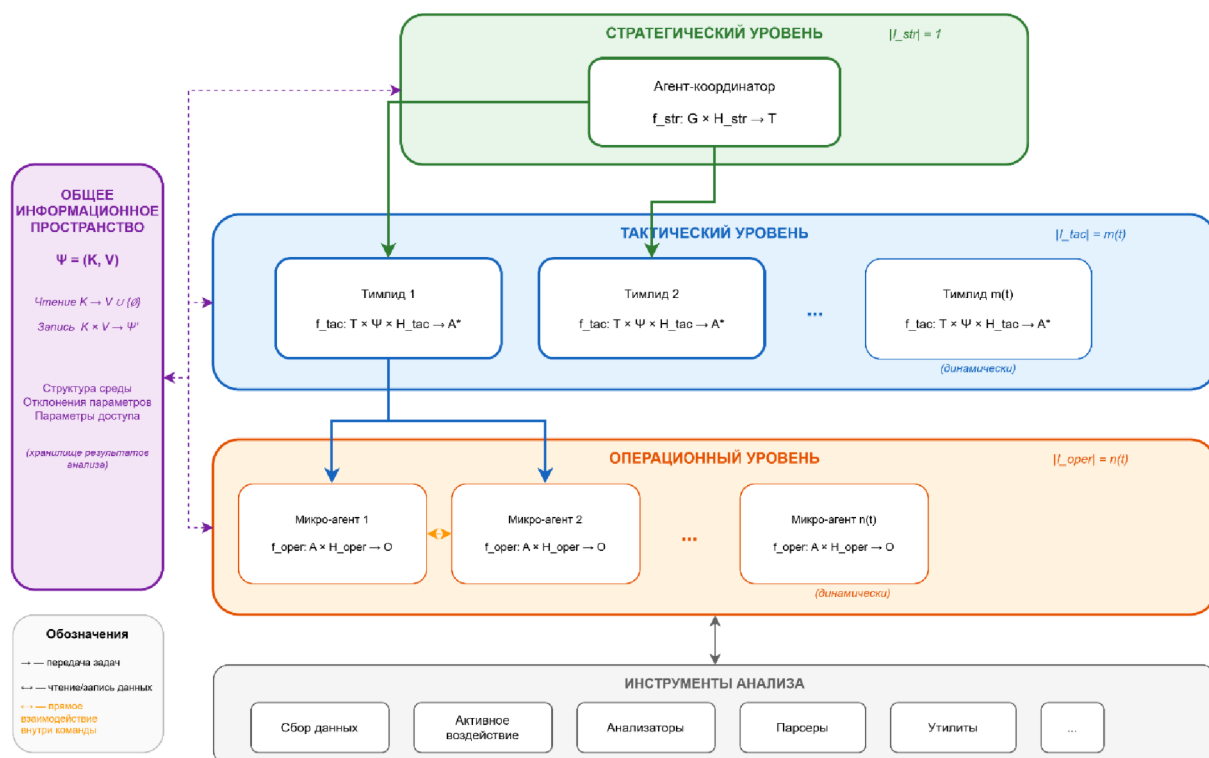


Рисунок 1. Трехуровневая архитектура многоагентной интеллектуальной системы (разработано авторами)

Figure 1. Three-level architecture of the multi-agent intelligent system (developed by the authors)

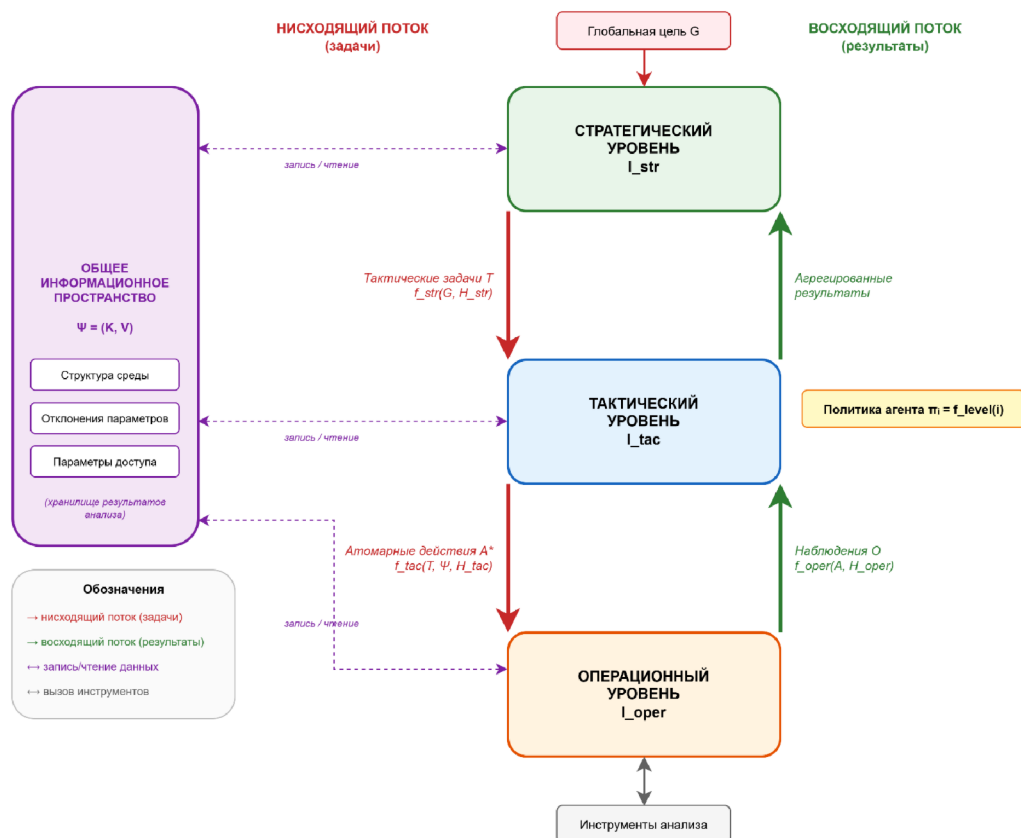


Рисунок 2. Поток данных между уровнями архитектуры и общее информационное пространство (разработано авторами)

Figure 2. Data flows between architecture levels and shared information space (developed by the authors)

Каждый уровень архитектуры обладает собственным набором доступных действий. Агент стратегического уровня создает и удаляет тимлидов, формулирует и назначает им задачи, однако не имеет прямого доступа к инструментам исследования. Агенты тактического уровня создают микро-агентов, декомпозируют задачи на атомарные операции и агрегируют результаты, но также не взаимодействуют с инструментами напрямую. Только агенты операционного

уровня выполняют непосредственные вызовы инструментов анализа. Такое разделение обеспечивает специализацию агентов и предотвращает выполнение нехарактерных для уровня действий.

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

Таблица 2. Сравнение архитектур многоагентных систем для автоматизированного анализа
Table 2. Comparison of multi-agent system architectures for automated analysis

Характеристика	Одноагентная система	Плоская многоагентная	Предлагаемая иерархическая
Число уровней	1	1	3
Специализация агентов	Отсутствует	Фиксированная	Динамическая по уровням
Координация	Не требуется	Прямой обмен сообщениями	Иерархическая с прямым взаимодействием внутри команд и общее информационное пространство
Масштабируемость	Низкая	Средняя	Высокая
Декомпозиция задач	Отсутствует	Горизонтальная	Иерархическая
Эмерджентность поведения	Отсутствует	Ограничена	Обеспечивается

Источник: разработано авторами
Source: developed by the authors

Одноагентные системы реализуют всю функциональность в рамках единого агента, что ограничивает возможности специализации и масштабирования. Плоские многоагентные системы распределяют функции между равноправными агентами, однако отсутствие иерархии затрудняет декомпозицию сложных задач и координацию при увеличении числа агентов [8, 9, 10]. Предлагаемая архитектура сочетает иерархическую декомпозицию задач между уровнями с прямым взаимодействием агентов внутри команд, что обеспечивает структурированное управление сложностью и гибкость оперативного взаимодействия.

Таким образом, предлагаемая авторская архитектура обеспечивает иерархическую декомпозицию задачи автоматизированного анализа на три уровня. Стратегический уровень осуществляет глобальное планирование и распределение задач между направлениями исследования. Тактический уровень обеспечивает специализированное управление в рамках отдельных направлений и формирование последовательностей атомарных операций. Операционный уровень выполняет непосредственное взаимодействие с инструментами и исследуемой средой. Динамическое формирование состава тактического и операционного

уровней обеспечивает адаптацию системы к текущим потребностям. Сочетание иерархической координации с автономностью агентов создает условия для эмерджентного поведения, когда коллективные стратегии системы превосходят возможности отдельных агентов.

ЗАКЛЮЧЕНИЕ

Разработанная формальная модель на основе расширения аппарата децентрализованных марковских процессов принятия решений в условиях частичной наблюдаемости позволяет учитывать три существенные особенности задачи автоматизированного анализа распределенных программно-аппаратных комплексов, которые не находят отражения в классической постановке. Композитное пространство состояний явно разделяет параметры исследуемой среды и накопленные агентами сведения. Динамическое пространство действий формализует расширение возможностей агентов по мере получения новой информации. Механизм верификации обеспечивает корректность обратной связи в условиях разреженного вознаграждения.

Предложенная трехуровневая иерархическая архитектура многоагентной интеллектуальной системы, в отличие от одноагентных и плос-

ких многоагентных решений, обеспечивает декомпозицию задачи на уровни с различной степенью абстракции и динамическое формирование состава агентов в процессе функционирования. Формальная связь архитектуры с оптимизационной моделью через понятие политики агента создает теоретическую основу для применения методов обучения с подкреплением на каждом уровне иерархии.

Вместе с тем представленные методологические основы определяют структуру и математический аппарат системы, однако не затрагивают вопросы оценки качества формируемых поведенческих стратегий и механизмы обучения агентов. Дальнейшее исследование предполагает разработку системы критериев оценки каче-

ства поведенческих стратегий агентов, включающей такие характеристики, как полнота покрытия, достоверность идентификации, результативность, неинвазивность и устойчивость, а также многокомпонентной функции полезности с адаптивными весовыми коэффициентами для их агрегирования. Другим направлением является разработка метода кросс-уровневого обучения агентов, обеспечивающего координированное формирование моделей всех уровней иерархии на основе анализа совместных траекторий выполнения задач.

*Авторы заявляют об отсутствии
конфликта интересов.*

The authors declare no conflict of interest.

ЛИТЕРАТУРА

1. Нигматуллин Р.Р. Системный анализ как направление научных исследований. *Вестник Российской академии наук*. 2007. Т. 77. № 2. С. 101–107.
2. 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. DOI: 10.6060/snt.20258404.00014.
3. 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. DOI: 10.6060/ivecofin.2023573.657.
4. 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. DOI: 10.6060/snt.20237604.0008.
5. Бобков С.П., Астраханцев Р.Г., Павлова Е.А. Исследование структуры потоков в технологических аппаратах с использованием дискретных динамических моделей. *Современные наукоемкие технологии. Региональное приложение*. 2024. № 1(77). С. 95–101. DOI: 10.6060/snt.20247701.00013.
6. ГОСТ Р 59277–2020. Информационные технологии. Многоагентные системы. Термины и определения. Введ. 2020-09-01. М.: Стандартинформ. 2020. 12 с.
7. Dorri A., Kanhere S.S., Jurdak R. Multi-Agent Systems. *IEEE Access*. 2018. Vol. 6. P. 28573–28593. DOI: 10.1109/ACCESS.2018.2831228.
8. 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. DOI: 10.17308/sait/1995-5499/2024/4/167-190.
9. Jennings N.R., Wooldridge M. Agent-Oriented Software Engineering. *Artificial Intelligence*. 2000. Vol. 117. P. 277–296.
10. D'Aniello G., De Falco M., Mastrandrea N. Designing a multi-agent system architecture for managing distributed operations within cloud manufacturing. *Evolutionary Intelligence*. 2020. DOI: 10.1007/s12065-020-00390-z.
11. Kaelbling L.P., Littman M.L., Cassandra A.R. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*. 1998. Vol. 101. N 1-2. P. 99–134. DOI: 10.1016/S0004-3702(98)00023-X.
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. DOI: 10.1287/moor.27.4.819.297.

REFERENCES

1. Nigmatullin R.R. Systems analysis as a research direction. *Bulletin of the Russian Academy of Sciences*. 2007. Vol. 77. N 2. P. 101–107. (in Russian).
2. Astrakhantseva I.A., Astrakhantsev R.G., Bobkov S.P. Agent-based intelligent systems from the point of view of systems analysis. *Modern High Technologies. Regional Application*. 2025. N 4(84). P. 101–109. DOI: 10.6060/snt.20258404.00014. (in Russian).
3. Astrakhantseva I.A., Gorev S.V., Astrakhantsev R.G. Systems approach to the analysis of the fractal nature of complex technical systems. *Ivecofin*. 2023. N 3(57). P. 89–97. DOI: 10.6060/ivecofin.2023573.657. (in Russian).
4. Astrakhantseva I.A., Gorev S.V., Astrakhantsev R.G. Fractal-based analysis of efficiency and reliability in complex technical systems. *Modern High Technologies. Regional Application*. 2023. N 4(76). P. 60–68. DOI: 10.6060/snt.20237604.0008. (in Russian).
5. Bobkov S.P., Astrakhantsev R.G., Pavlova E.A. Research of the structure of flow in technological equipment using discrete dynamic models. *Modern High Technologies. Regional Application*. 2024. N 1(77). P. 95–101. DOI: 10.6060/snt.20247701.00013. (in Russian).
6. GOST R 59277–2020. Information Technologies. Multi-Agent Systems. Terms and Definitions. Moscow: Standartinform. 2020. 12 p. (in Russian).
7. Dorri A., Kanhere S.S., Jurdak R. Multi-Agent Systems: *IEEE Access*. 2018. Vol. 6. P. 28573–28593. DOI: 10.1109/ACCESS.2018.2831228.
8. Prinev M.A., Ledeneva T.M., Garshina V.V. Multi-agent systems: a review of modern approaches to modeling and design (Part 2). *Bulletin of VGU. Series: Systems Analysis and Information Technologies*. 2024. N 4. P. 167–190. DOI: 10.17308/sait/1995-5499/2024/4/167-190. (in Russian).
9. Jennings N.R., Wooldridge M. Agent-Oriented Software Engineering. *Artificial Intelligence*. 2000. Vol. 117. P. 277–296.
10. D'Aniello G., De Falco M., Mastrandrea N. Designing a multi-agent system architecture for managing distributed operations within cloud manufacturing. *Evolutionary Intelligence*. 2020. DOI: 10.1007/s12065-020-00390-z.
11. Kaelbling L.P., Littman M.L., Cassandra A.R. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*. 1998. Vol. 101. N 1-2. P. 99–134. DOI: 10.1016/S0004-3702(98)00023-X.
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. DOI: 10.1287/moor.27.4.819.297.

13. **Oliehoek F.A., Spaan M.T.J., Vlassis N.** A Concise Introduction to Decentralized POMDPs. *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (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: Applications and Reviews*. 2008. Vol. 38. N 2. P. 156–172. DOI: 10.1109/TSMCC.2007.913919.
15. **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. DOI: 10.1088/1742-6596/2001/1/012002.
16. **Koops W., Junges S., Jansen N.** Approximate Dec-POMDP Solving Using Multi-Agent A*. *Proceedings of the 33rd International Conference on Automated Planning and Scheduling (ICAPS 2024)*. 2024. P. 745–753.
17. **Бобков С.П., Галиаскаров Э.Г.** Моделирование процесса теплопроводности с использованием систем клеточных автоматов. *Программные продукты и системы*. 2020. № 4. С. 641–650. DOI: 10.15827/0236-235X.132.641-650.
18. **Бобков С.П., Астраханцев Р.Г., Самарский А.А., Павлова Е.А.** Имитационная модель движения вещества в технологическом аппарате. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2024. № 2(60). С. 32–40. DOI: 10.6060/ivecofin.2024602.681.
19. **Бобков С.П., Астраханцева И.А., Гушин А.А. и др.** Использование дискретного вероятностного подхода для моделирования проточных трубчатых реакторов. *Известия высших учебных заведений. Серия: Химия и химическая технология*. 2025. Т. 68. № 2. С. 96–101. DOI: 10.6060/ivkkt.20256802.7109.
20. **Самсонова А.И., Суворов И.А., Астраханцев Р.Г.** Интеллектуальная система анализа пользовательской активности в защищенных операционных средах. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2025. № 1(63). С. 79–85. DOI: 10.6060/ivecofin.2025631.715.
21. **Бобков С.П., Астраханцева И.А., Павлова Е.А.** Имитационное моделирование для интеллектуальной поддержки принятия управленческих решений. *Современные наукоемкие технологии. Региональное приложение*. 2022. № 1(69). С. 61–69. DOI: 10.6060/snt.20226901.0008.
22. **Зимнуров М.Ф., Астраханцева И.А., Гримицкий П.Н.** Системный анализ и оптимизация количественных показателей эффективности в технологических проектах на основе гибких методологий. *Современные наукоемкие технологии. Региональное приложение*. 2023. № 3(75). С. 61–68. DOI: 10.6060/snt.20237503.0008.
13. **Oliehoek F.A., Spaan M.T.J., Vlassis N.** A Concise Introduction to Decentralized POMDPs. *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (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: Applications and Reviews*. 2008. Vol. 38. N 2. P. 156–172. DOI: 10.1109/TSMCC.2007.913919.
15. **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. DOI: 10.1088/1742-6596/2001/1/012002.
16. **Koops W., Junges S., Jansen N.** Approximate Dec-POMDP Solving Using Multi-Agent A*. *Proceedings of the 33rd International Conference on Automated Planning and Scheduling (ICAPS 2024)*. 2024. P. 745–753.
17. **Bobkov S.P., Galiaskarov E.G.** Simulation of the heat conduction process using cellular automata systems. *Software & Systems*. 2020. N 4. P. 641–650. DOI: 10.15827/0236-235X.132.641-650. (in Russian).
18. **Bobkov S.P., Astrakhantsev R.G., Samarskiy A.A., Pavlova E.A.** Simulation model of the movement of substance in a technological apparatus. *Ivecofin*. 2024. N 2(60). P. 32–40. DOI: 10.6060/ivecofin.2024602.681. (in Russian).
19. **Bobkov S.P., Astrakhantseva I.A., Gushchin A.A. et al.** Using a discrete probabilistic approach for simulating flow tubular reactors. *ChemChemTech*. 2025. Vol. 68. N 2. P. 96–101. DOI: 10.6060/ivkkt.20256802.7109. (in Russian).
20. **Samsonova A.I., Suvorov I.A., Astrakhantsev R.G.** Intelligent system for analyzing user activities in secure operating environments. *Ivecofin*. 2025, N 1(63). P. 79–85. DOI: 10.6060/ivecofin.2025631.715. (in Russian).
21. **Bobkov S.P., Astrakhantseva I.A., Pavlova E.A.** Simulation modeling for intellectual support of management decision-making. *Modern High Technologies. Regional Application*. 2022. N 1(69). P. 61–69. DOI: 10.6060/snt.20226901.0008. (in Russian).
22. **Zimnurov M.F., Astrakhantseva I.A., Grimenitsky P.N.** Systems analysis and optimization of quantitative performance indicators in technological projects based on flexible methodologies. *Modern High Technologies. Regional Application*. 2023. N 3(75). P. 61–68. DOI: 10.6060/snt.20237503.0008. (in Russian).

Поступила в редакцию 02.10.2025
Принята к опубликованию 16.10.2025

Received 02.10.2025
Accepted 16.10.2025