

УДК 004.89
<https://doi.org/10.32362/2500-316X-2023-11-4-16-25>



НАУЧНАЯ СТАТЬЯ

О подходе к управлению знаниями и разработке мультиагентной системы представления и обработки знаний

Е.И. Зайцев[@],
Е.В. Нурматова

МИРЭА – Российский технологический университет, Москва, 119454 Россия
[@] Автор для переписки, e-mail: zajcev@mirea.ru

Резюме

Цели. Мультиагентная система представления и обработки знаний (МСПОЗ) – это распределенная система искусственного интеллекта, предназначенная для решения проблем, которые трудно или невозможно решить с помощью монолитной интеллектуальной системы. Решение сложных проблем в МСПОЗ осуществляется интеллектуальными программными агентами, которые инкапсулируют в программных классах когнитивные структуры данных, методы логического вывода и машинного обучения. Интеллектуальные программные агенты МСПОЗ способны рационально действовать в условиях неполноты и нечеткости поступающей информации. Целями работы являются исследование и разработка моделей, методов, программных модулей и инструментальных программных средств, которые позволяют создать высокоэффективную МСПОЗ.

Методы. В работе использовались методы агентного моделирования, позволяющие формально описывать и программно имитировать рациональное поведение интеллектуальных агентов, методы экспертных оценок, математический аппарат теории автоматов, марковские цепи, нечеткая логика, нейронные сети, алгоритмы машинного обучения с подкреплением.

Результаты. Разработаны структурная схема МСПОЗ, мультиагентный решатель, схема управления доступом к микросервисам. Предложены методы распределения интеллектуальных программных агентов по узлам МСПОЗ, а также алгоритмы оптимизации логической структуры распределенной базы знаний (РБЗ), позволяющие повысить эффективность объемных, стоимостных и временных характеристик МСПОЗ.

Выводы. Предложен подход к разработке и использованию интеллектуальных программных агентов, который объединяет механизмы рассуждений на основе знаний с нейросетевыми моделями. Разработаны структура МСПОЗ, схема управления РБЗ, методы оптимизации РБЗ, определения доступности используемых агентами микросервисов, обеспечения надежности и скоординированного функционирования вычислительных узлов системы, а также инструментальные программные средства, позволяющие упростить процесс проектирования и реализации МСПОЗ. Полученные результаты демонстрируют эффективность представленного подхода к управлению знаниями и разработке высокопроизводительной проблемно-ориентированной МСПОЗ.

Ключевые слова: мультиагентная система, интеллектуальные программные агенты, мультиагентный интеллектуальный решатель, система представления и обработки знаний, обучение с подкреплением

• Поступила: 24.10.2022 • Доработана: 27.01.2023 • Принята к опубликованию: 02.05.2023

Для цитирования: Зайцев Е.И., Нурматова Е.В. О подходе к управлению знаниями и разработке мультиагентной системы представления и обработки знаний. *Russ. Technol. J.* 2023;11(4):16–25. <https://doi.org/10.32362/2500-316X-2023-11-4-16-25>

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

Авторы заявляют об отсутствии конфликта интересов.

RESEARCH ARTICLE

Approach to knowledge management and the development of a multi-agent knowledge representation and processing system

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

MIREA – Russian Technological University, Moscow, 119454 Russia

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

Abstract

Objectives. A multi-agent knowledge representation and processing system (MKRPS) comprises a distributed artificial intelligence system designed to solve problems that are difficult or impossible to solve using monolithic systems. Solving complex problems in an MKRPS is accomplished by communities of intelligent software agents that use cognitive data structures, logical inference, and machine learning. Intelligent software agents are able to act rationally under conditions of incompleteness and ambiguity of incoming information. The aim of the present work is to identify models and methods, as well as software modules and tools, for use in developing a highly efficient MKRPS.

Methods. Agent-based modeling methods were used to formally describe and programmatically simulate the rational behavior of intelligent agents, expert evaluation methods, the mathematical apparatus of automata theory, Markov chains, fuzzy logic, neural networks, and reinforcement learning.

Results. An MKRPS structure diagram, a multi-agent solver, and microservices access control diagram were developed. Methods for distribution of intelligent software agents on the MKRPS nodes are proposed along with algorithms for optimizing the logical structure of the distributed knowledge base (DKB) to improve the performance of the MKRPS in terms of volume, cost and time criteria.

Conclusions. The proposed approach to the development and use of intelligent software agents combines knowledge-based reasoning mechanisms with neural network models. The developed MKRPS structure and DKB control diagram includes described methods for optimizing the DKB, determining the availability of microservices used by the agents, ensuring the reliability assurance and coordinated functioning of the computing nodes of the system, as well as instrumental software tools to simplify the design and implementation of the MKRPS. The results demonstrate the effectiveness of the presented approach to knowledge management and the development of a high-performance problem-oriented MKRPS.

Keywords: multi-agent system, intelligent software agents, multi-agent intelligent solver, knowledge representation and processing system, reinforcement learning

• Submitted: 24.10.2022 • Revised: 27.01.2023 • Accepted: 02.05.2023

For citation: Zaytsev E.I., Nurmatova E.V. Approach to knowledge management and the development of a multi-agent knowledge representation and processing system. *Russ. Technol. J.* 2023;11(4):16–25. <https://doi.org/10.32362/2500-316X-2023-11-4-16-25>

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

The authors declare no conflicts of interest.

ВВЕДЕНИЕ

В мультиагентной системе представления и обработки знаний (МСПОЗ) решение сложных плохо формализуемых проблем осуществляется интеллектуальными программными агентами, которые способны рационально действовать в условиях неопределенности, неполноты и нечеткости поступающей информации [1–5]. Принимая решения и осуществляя рациональные действия, программные агенты используют базу знаний и событийно-управляемые микросервисы, спроектированные как отдельные взаимодействующие процессы с легковесными межпроцессными связями. Взаимодействие агентов с микросервисами осуществляется через события, в качестве которых могут выступать как простые уведомления, так и сложные структуры с поддержкой состояний.

Решение сложных проблем в МСПОЗ осуществляется путем декомпозиции проблем на подзадачи, которые совместно решают реактивные и когнитивные прикладные программные агенты. Используется как горизонтальная декомпозиция, приводящая к созданию многосвязной системы с плоской структурой, так и вертикальная декомпозиция, которая создает иерархическую систему с несколькими уровнями.

Благодаря реализации в МСПОЗ методов машинного обучения с подкреплением (reinforcement learning, RL) поведение прикладных программных агентов становится более рациональным при многократном решении задач. Для обучения прикладных программных агентов в МСПОЗ используется алгоритм Actor-Critic [6–10].

С целью повышения производительности в МСПОЗ реализованы специализированные прикладные программные интерфейсы (application programming interface, API) и системные программные модули, ассоциированные с системными программными агентами. Системные программные агенты осуществляют планирование и управление вычислительными ресурсами МСПОЗ, а также обеспечивают мобильность прикладных программных агентов. Прикладные агенты могут перемещаться по узлам МСПОЗ вместе с необходимым для них окружением. В отличие от контейнеров, реализуемых на основе пространств имен (например, платформой

Docker¹), для поддержки технологии мобильных агентов в МСПОЗ используются специализированные модули LibOS (library operating system).

Производительность МСПОЗ во многом определяется выбранным способом структуризации, хранения и обработки знаний [11, 12]. Был разработан высокопроизводительный проблемно-ориентированный мультиагентный решатель, и оптимизирована логическая структура распределенной базы знаний (РБЗ), имеющая минимальное общее время обработки запросов и транзакций.

СТРУКТУРА МСПОЗ

Структурная схема МСПОЗ изображена на рис. 1. На каждом вычислительном узле МСПОЗ функционируют прикладные (интеллектуальные) программные агенты двух типов: реактивные и когнитивные [13]. Прикладные программные агенты взаимодействуют между собой, а также с системными программными агентами, входящими в состав внешней разделяемой библиотеки пользовательского уровня LibOS, которая ориентирована на экзоядерную архитектуру операционной системы (ОС). Системные программные агенты используются для планирования и управления вычислительными ресурсами, распределения нагрузки и мониторинга системы.

Для представления агентно-ориентированных абстракций (целей, желаний, намерений, убеждений агентов) и реализации логического вывода используются специальные программные методы и когнитивные структуры данных (cognitive data structures, CDS), ассоциированные с когнитивными программными агентами.

На рис. 2 показан пример диаграммы состояний и переходов когнитивного программного агента одного из узлов МСПОЗ.

Как следует из диаграммы, когнитивный программный агент может находиться в одном из пяти состояний, два из которых являются составными (composite state), т.е. имеют вложенные состояния. Изменение состояния программного агента происходит в результате некоторого события. Возможен

¹ <https://www.docker.com/>. Дата обращения 20.03.2023. / Accessed March 20, 2023.

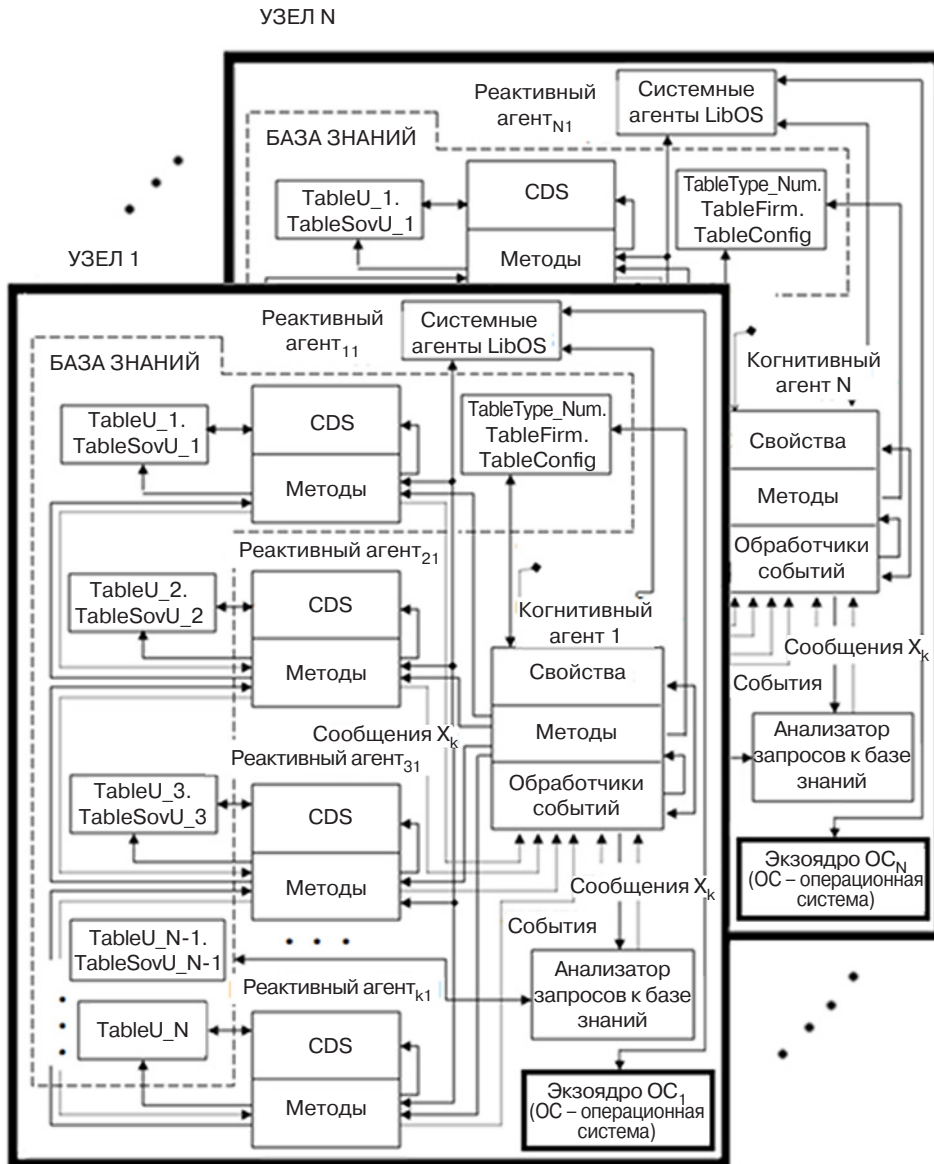


Рис. 1. Структура МСПОЗ

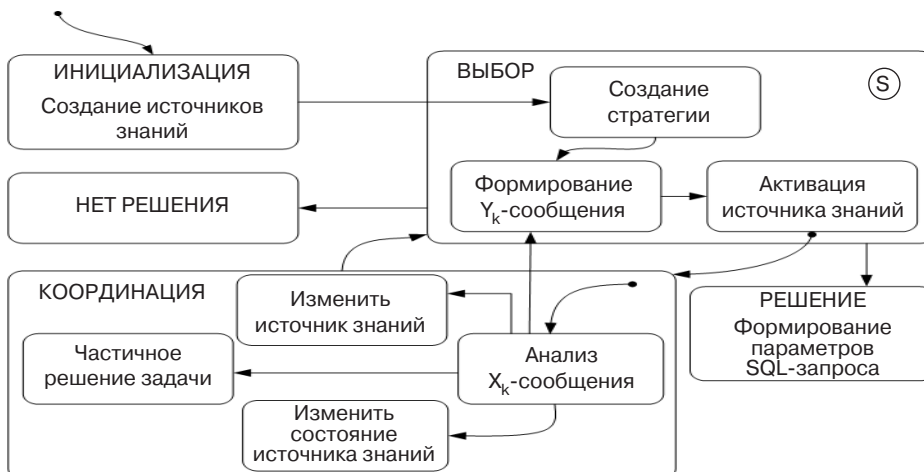


Рис. 2. Диаграмма состояний и переходов когнитивного агента

переход в новое состояние без совершения события, который осуществляется сразу после выполнения действий (или активностей), связанных с предыдущим состоянием.

Из состояния «Инициализация» когнитивный агент переходит в составное состояние «Выбор». В этом состоянии генерируется стратегия и выбирается необходимый источник знаний с учетом установленных на этапе инициализации связей, а также поступающих от других программных агентов осведомительных сообщений. Далее когнитивный программный агент переходит в составное состояние «Координация», в котором активизируются новые источники знаний и координируются действия реактивных программных агентов.

Если реактивные программные агенты не находят согласованного решения (состояние «Нет решения»), когнитивный программный агент возвращается во вложенное состояние частичного решения задачи. Если решение найдено (состояние «Решение»), то полученные на данном этапе данные используются для формирования новых запросов к базе знаний.

Для работы с базой знаний реализованы четыре типа методов, формирующих запросы и обрабатывающих результаты этих запросов:

- анализ (метод ANS) – реализует логический анализ событий;
- ассоциация (метод ASS) – используется для получения ответов на запросы о связях между объектами и событиями;
- сравнение событий или объектов (метод CMP);
- спецификация объектов (метод VAL).

Для спецификации объектов могут использоваться как четкие, так и нечеткие запросы к базе

знаний. При реализации нечетких запросов могут использоваться различные типы функций принадлежности, которые выбираются инженером по знаниям в результате вычислительных экспериментов.

Когнитивные программные агенты координируют работу реактивных программных агентов, ассоциированных с локальными источниками знаний. Пример диаграммы взаимодействия реактивных программных агентов одного из узлов МСПОЗ представлен на рис. 3.

На диаграмме взаимодействия время движется сверху вниз, программные компоненты доступа к базе знаний и агенты представлены вертикальными линиями. Сообщения между агентами (компонентами) отмечены горизонтальными стрелками. При получении сообщения каждый агент (источник знаний) осуществляет вызов соответствующего программного метода (функции-члена класса $F()$ или $S()$) и возвращает результат. Слева на рисунке приведены комментарии. В данном случае каждый реактивный программный агент с номером k взаимодействует только со своим ближайшим соседом с номером $(k - 1)$. Программные агенты этой группы последовательно выполняют возложенные на них задачи в рамках одного процесса (без переключения контекста). Приоритеты реактивных программных агентов узла МСПОЗ устанавливаются в соответствии с порядковым номером агента. Первый реактивный агент использует высокоприоритетные фреймы, связанные с таблицами TableU_1 и TableSovU_1. Программный агент с номером k имеет самый низкий приоритет и связан с таблицей TableU_N.

Если узлом МСПОЗ является мультипроцессорная система, то агенты этого узла могут действовать одновременно. Для организации параллельных

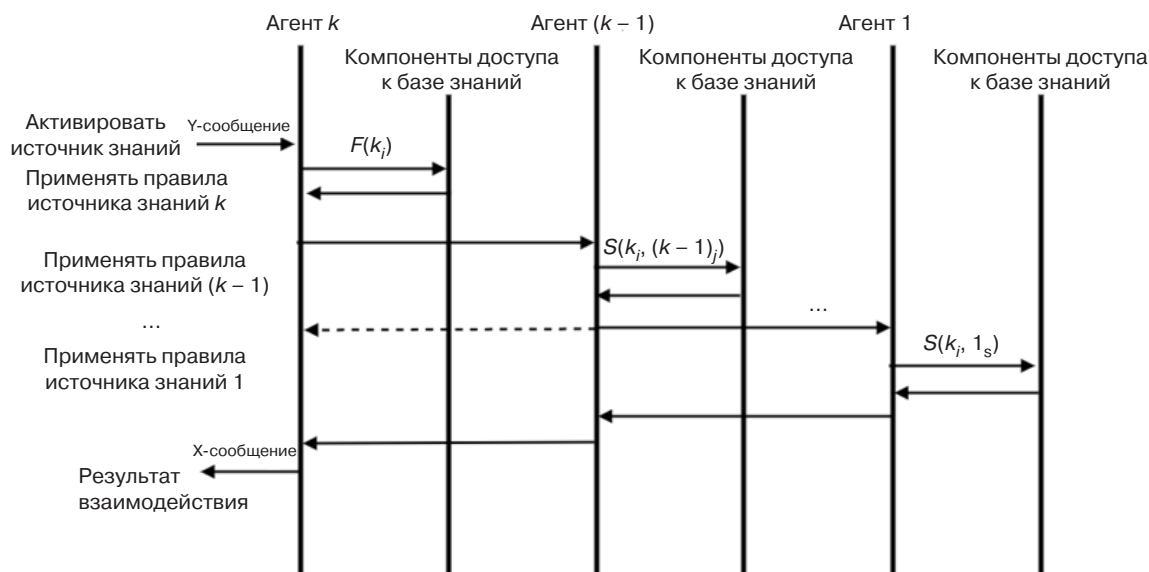


Рис. 3. Схема взаимодействия реактивных программных агентов

вычислений прикладные программные агенты каждого узла МСПОЗ распределяются по группам с использованием матриц совместимости и включения.

Матрица совместимости S имеет следующий вид:

$$S = \begin{bmatrix} 0 & s_{12} & s_{13} & \dots & s_{1M} \\ s_{21} & 0 & & \dots & s_{2M} \\ s_{31} & s_{32} & 0 & \dots & s_{3M} \\ \dots & \dots & \dots & \dots & \dots \\ s_{M1} & s_{M2} & s_{M3} & \dots & 0 \end{bmatrix} \begin{matrix} S_1 \\ S_2 \\ S_3 \\ \dots \\ S_M \end{matrix},$$

где $s_{ij} = 1$, если агенты A_i и A_j должны работать параллельно, иначе $s_{ij} = 0$; S_i – i -я строка матрицы совместимости S ; M – число агентов.

Для распределения программных агентов узла по группам используется матрица включения R :

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1M} \\ r_{21} & r_{22} & \dots & r_{2M} \\ \dots & \dots & \dots & \dots \\ r_{H1} & r_{H2} & \dots & r_{HM} \end{bmatrix} \begin{matrix} R_1 \\ R_2 \\ \dots \\ R_H \end{matrix},$$

где H – число групп; $r_{ij} = 1$, если агент A_i включен в группу Y_j . Агент A_i входит в группу Y_j , если $S_i \cap R_j = \emptyset$, т.е. строки матрицы не пересекаются.

Для оптимального разбиения множества агентов на подмножества при использовании матриц совместимости и включения необходимо учитывать структуру узла МСПОЗ, функциональные особенности программных агентов и их требования к вычислительным ресурсам.

В мультиагентном обучении с подкреплением (multi-agent reinforcement learning, MARL) окружающая среда зависит от всех программных агентов. В отличие от централизованного обучения, при котором программные агенты полностью контролируют вычислительный процесс, при этом политики распределяются центральным агентом, в децентрализованной модели, используемой в МСПОЗ, агенты независимы и могут обмениваться опытом и политиками. В децентрализованной модели выполнение и обучение реализуются локально, что позволяет прикладным программным агентам адаптироваться к местному восприятию окружающей среды (рис. 4).

Децентрализованная модель мультиагентного обучения с подкреплением позволяет использовать стандартные алгоритмы RL. Прикладные программные агенты МСПОЗ обучаются через серию поощрений и наказаний на основе алгоритма Actor-Critic, в котором стратегия генерирует действия, а функция ценности критикует эти действия.

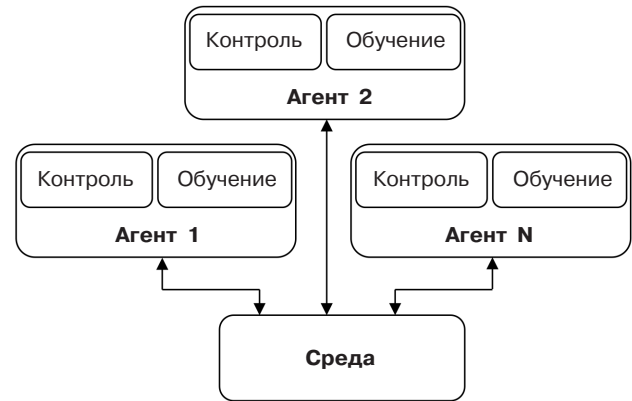


Рис. 4. Децентрализованная модель мультиагентного обучения с подкреплением

Поскольку для каждого программного агента имеются актор и критик, агенты могут иметь разные стратегии (политики) и вознаграждения. Агенты МСПОЗ сотрудничают для оптимизации общей долгосрочной цели. Аппроксимация функций в МСПОЗ реализуется на базе нейронных сетей, которые моделируют как функции политики, так и функции ценности.

Для эффективной реализации мультиагентного обучения с подкреплением необходимо решить проблемы многомерных и мультимодальных целей, масштабируемости, нестабильности и оптимальности [7].

В процессе решения подзадач программные агенты используют микросервисы, которые для повышения надежности и производительности дублируются на разных узлах МСПОЗ. Системные программные агенты распределяют вычислительную нагрузку и управляют микросервисами на основе данных, которые предоставляют агенты-мониторы (рис. 5).

СИНТЕЗ РАСПРЕДЕЛЕННЫХ КОГНИТИВНЫХ СТРУКТУР ДАННЫХ

Выполнение операций со структурами данных в МСПОЗ, будь то пополнение, либо извлечение данных, осуществляется по ключам вследствие большой размерности создаваемых когнитивных структур данных. Для этого вся логическая структура CDS должна быть разбита на ряд кластеров, которые имеют наименьшую взаимосвязь при различных ограничениях. В данном случае примем ограничения на размерность кластеров и ограничения на степень семантической близости логических записей, входящих в кластеры. При этом будет учитываться тип используемых систем хранения данных.

Введем бинарный параметр Z_{kj}^i , который характеризует использование k -м запросом i -й группы данных, отнесенной к j -й логической записи. Расчет данного параметра основывается на бинарной переменной a_{ik} , равной единице, если i -я группа данных

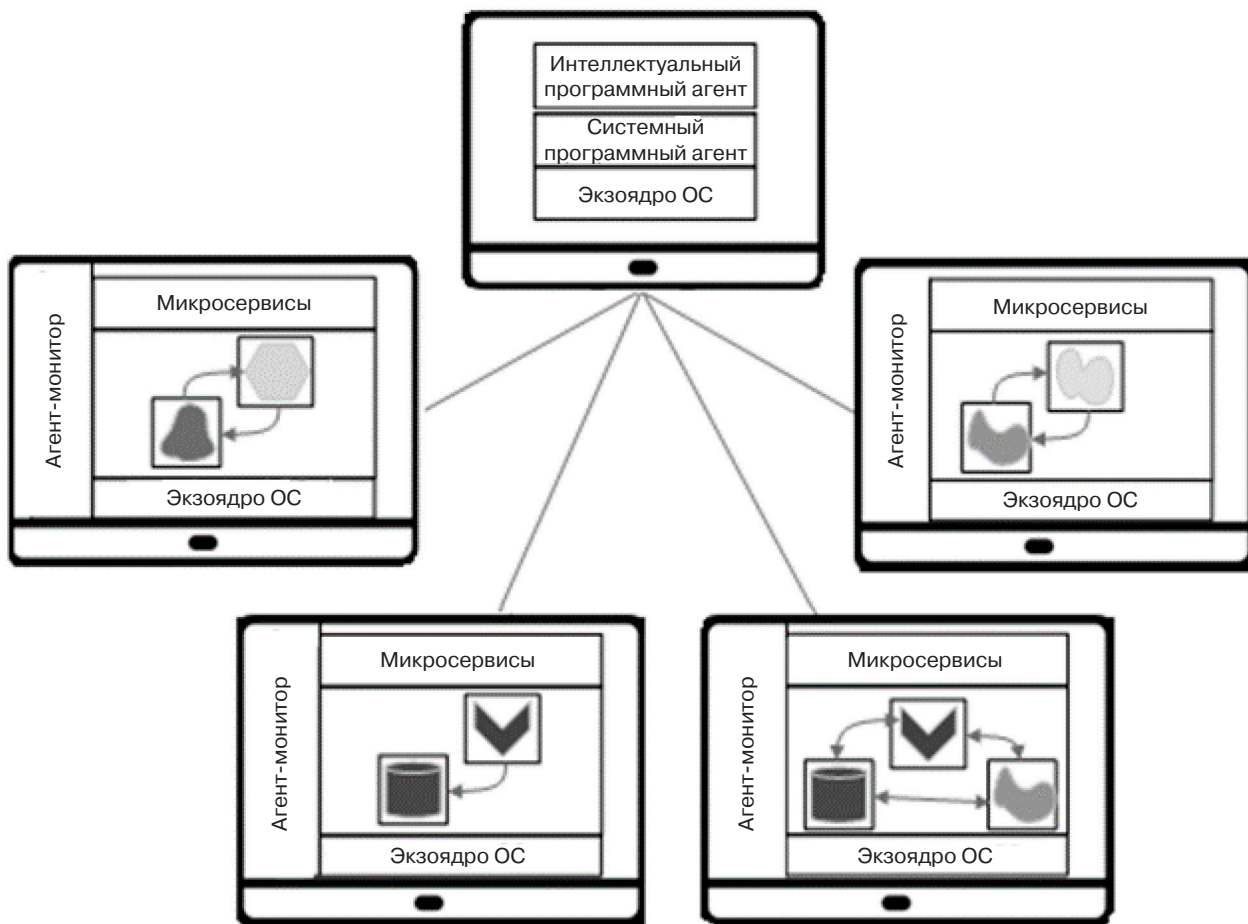


Рис. 5. Микросервисы и системные агенты-мониторы

включена в k -й запрос, и нулю – в противном случае. Учитывая по логике связи, что группы данных входят в логические записи, этот параметр усилен переменной x_{ij} , которая по аналогии равна единице, если i -я группа включена в j -ю логическую запись; $x_{ij} = 0$ – в противном случае.

Имеем следующий расчет параметра Z_{kj}^i :

$$Z_{kj}^i = \begin{cases} 1, & \text{при } \sum_{i=1}^I a_{ik} x_{ij} \geq 1, \\ 0, & \text{при } \sum_{i=1}^I a_{ik} x_{ij} = 0. \end{cases}$$

Полную задачу синтеза распределенной структуры данных для МСПОЗ будем решать с учетом критерия минимального суммарного времени исполнения запросов пользователей при таких ограничениях, как уникальность включения наборов данных в запись, длина логической записи, общее число типов логических записей в структуре, время поиска данных по запросам, уникальность входных узлов в структуре и их общее число.

В данной работе приведем приближенный алгоритм решения задачи синтеза оптимальной распределенной структуры данных по критерию минимального общего времени выполнения запроса. Итак, определяем распределение групп по критерию минимума суммарного трафика, для чего используем приближенный алгоритм распределения кластеров данных между сервером и клиентами локальной сети. И далее сводим канонический граф структуры данных к несвязному графу с вычислением веса каждой группы данных.

Вес группы данных включает веса самой группы и веса дуг с учетом требований пользователей МСПОЗ:

$$V_i = V_i^{\text{ГП}} + V_{i'i'}^{\text{СВ}},$$

где $V_i^{\text{ГП}}$ – общий вес группы данных; $V_{i'i'}^{\text{СВ}}$ – вес дуг связанного графа канонической структуры данных; i' – индекс группы, смежной с i -й группой данных.

$$V_i^{\text{ГП}} = \sum_{k=1}^{k_0} \sum_{p=1}^{p_0} \gamma_{kp}^3 \delta_{kp}^3 \vartheta_{pi},$$

$$V_{ii'}^{CB} = \sum_{k=1}^{k_0} \sum_{p=1}^{p_0} \gamma_{kp}^3 \delta_{kp}^3 \vartheta_{pi} \sum_{i' \neq i} \vartheta_{pi'} a_{ii'}^\Gamma,$$

где γ_{kp}^3 – частота использования запросов пользователями; δ_{kp}^3 – элементы матрицы запросов пользователей; $\vartheta_{pi'}$ – матрица групп данных при исполнении запросов; $a_{ii'}^\Gamma$ – матрица семантической смежности i -й группы данных с группой данных, имеющей индекс i' .

Для конкретной i -й группы вес будет составлять:

$$V_i = \sum_{k=1}^{k_0} \sum_{p=1}^{p_0} \gamma_{kp}^3 \delta_{kp}^3 \vartheta_{pi} \left(1 + \sum_{i' \neq i} \vartheta_{pi'} a_{ii'}^\Gamma \right).$$

Далее граф компьютерной сети преобразуем в несвязный граф с вычислением веса узла:

$$V_r = t_r + \sum_{r' \neq r}^{R_0} t_{rr'},$$

где t_r – общая средняя продолжительность обработки данных в r -м узле, состоящая из времени декомпозиции запроса на подзапросы, выбора маршрута, установления соединения; $t_{rr'}$ – средняя продолжительность передачи данных между узлами, определяемая на основе матрицы логических расстояний между серверами узлов компьютерной сети.

Далее формируется матрица $\mathbf{V} = \|v_{ir}\|$, элементы которой определены как декартово произведение веса каждого узла на вес каждой группы данных:

$$v_{ir} = V_i \times V_r \text{ для } i = \overline{1, I}, \quad r = \overline{1, R_0}.$$

После этого проблему

$$\min_{\{x_{ir}\}} \sum_{i=1}^I \sum_{r=1}^{R_0} v_{ir} x_{ir}$$

решаем со следующими ограничениями:

- количество групп данных, локализация которых возможна на одном узле:

$$\sum_{i=1}^I x_{ir} \leq N_r, \quad r = \overline{1, R_0};$$

- допустимая избыточность групп по узлам сети:

$$\sum_{r=1}^{R_0} x_{ir} \leq M_i, \quad i = \overline{1, I};$$

- объем доступной внешней памяти системы хранения данных:

$$\sum_{i=1}^I x_{ir} \rho_i \pi_i \leq \eta_r^{BZY},$$

где ρ_i – значения длин групп данных; π_i – значения числа экземпляров в группах; η_r^{BZY} – объем доступной памяти внешнего запоминающего устройства (ВЗУ) на узле; $x_{ir} = 1$, если i -я группа данных включена в r -й сетевой узел, иначе $x_{ir} = 0$.

Решение линейной задачи целочисленного программирования позволяет определить оптимальную локализацию групп данных по узлам сети, и, в дополнение, позволяет определить оптимальную структуру групп данных, размещенных по узлам сети.

На следующем этапе в каждом узле сети решаем задачу оптимального распределения групп данных узла по типам логических записей по критерию минимума общего времени локальной обработки данных. Количество задач синтеза здесь определяется количеством узлов сети. Исходными данными являются подграфы графа канонической структуры данных, а также временные и объемные характеристики подграфов их канонической структуры, набор запросов пользователей и узлов сети [14, 15]. Задача синтеза решается с помощью приближенных алгоритмов с ограничениями на количество групп в записи, на уникальность включения групп в запись, на стоимость хранения информации, на общее время обслуживания запроса. В результате определяются логические структуры базы данных для каждого узла сети, формируются матрицы распределения множества групп данных по типам логических записей, а затем – матрицы распределения множества групп записей по узлам сети.

Результаты решения задачи синтеза распределенных когнитивных структур данных имеют важное практическое значение для проектирования оптимальной структуры РБЗ и возможности формирования спецификаций для запросов и корректировки распределенных данных.

ЗАКЛЮЧЕНИЕ

В работе представлен методологический подход к разработке МСПОЗ. Описаны функциональная и структурная организации мультиагентного решателя, модели прикладных и системных программных агентов, методика проектирования и управления РБЗ. Рассмотрены методы распределения программных агентов по узлам МСПОЗ, определения доступности микросервисов, обеспечения надежной и согласованной работы вычислительных узлов. Приведены примеры диаграмм взаимодействия, состояний и переходов когнитивных и реактивных программных агентов МСПОЗ. Рассмотрены алгоритмы оптимизации логической структуры РБЗ. Оптимизация РБЗ позволяет повысить эффективность МСПОЗ по временным,

объемным и стоимостным характеристикам, делает ее более производительной, гибкой и функциональной. Полученные результаты подтверждают эффективность представленного подхода к разработке МСПОЗ.

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

Authors' contribution. All authors equally contributed to the research work.

СПИСОК ЛИТЕРАТУРЫ

1. Zaytsev E.I., Khalabiya R.F., Stepanova I.V., Bunina L.V. Multi-Agent System of Knowledge Representation and Processing. In: Kovalev S., Tarassov V., Snasel V., Sukhanov A. (Eds.). *Proceedings of the Fourth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'19). Advances in Intelligent Systems and Computing*. Springer; 2020. V. I. P. 131–141. https://doi.org/10.1007/978-3-030-50097-9_14
2. Baranauskas R., Janaviciute A., Jasinevicius R., Jukavicius V. On Multi-Agent Systems Intellectics. *Inf. Technol. Control*. 2015;44(1):112–121. <https://doi.org/10.5755/j01.itc.44.1.8768>
3. Darweesh S., Shehata H. Performance Evaluation of a Multi-Agent System using Fuzzy Model. *2018 First International Workshop on Deep and Representation Learning (IWDRL)*. 2018. P. 7–12. <https://doi.org/10.1109/IWDRL.2018.8358208>
4. Рассел С., Норвиг П. *Искусственный интеллект: современный подход*. Т. 2. Знания и рассуждения в условиях неопределенности: пер. с англ. СПб.: Диалектика; 2021. 480 с.
5. Рассел С., Норвиг П. *Искусственный интеллект: современный подход*. Т. 3. Обучение, восприятие и действие: пер. с англ. СПб.: Диалектика; 2022. 640 с.
6. Саттон Р.С., Барто Э.Дж. *Обучение с подкреплением*: пер. с англ. М.: ДМК Пресс; 2020. 552 с.
7. Грессер Л., Кенг В.Л. *Глубокое обучение с подкреплением: теория и практика на языке Python*: пер. с англ. СПб.: Питер; 2022. 416 с. [Graesser L., Keng W.L. *Foundations of Deep Reinforcement Learning: Theory and Practice in Python*. Addison-Wesley Professional; 2020. 416 p.]
8. Памперла М., Фергюсон К. *Глубокое обучение и игра в го*: пер. с англ. М.: ДМК Пресс; 2020. 372 с. [Pumperla M., Ferguson K. *Deep Learning and the Game of Go*. Manning; 2019. ISBN 978-1-6172-9532-4. 384 p.]
9. Редько В.Г. *Эволюция, нейронные сети, интеллект: Модели и концепции эволюционной кибернетики*. М.: ЛИБРОКОМ; 2013. 224 с.
10. Winder P. *Reinforcement Learning. Industrial Applications of Intelligent Agents*. O'Reilly Media, Inc.; 2021. 382 p.
11. Houhamdi Z., Athamena B., Abuzaineddin R., Muhairat M. A Multi-Agent System for Course Timetable Generation. *TEM Journal*. 2019;8(1):211–221. <https://doi.org/10.18421/TEM81-30>
12. Aly S., Badoor H. Performance Evaluation of a Multi-Agent System using Fuzzy Model. *2018 First International Workshop on Deep and Representation Learning (IWDRL)*. 2018. P. 175–189. <https://doi.org/10.1109/IWDRL.2018.8358208>

REFERENCES

1. Zaytsev E.I., Khalabiya R.F., Stepanova I.V., Bunina L.V. Multi-Agent System of Knowledge Representation and Processing. In: Kovalev S., Tarassov V., Snasel V., Sukhanov A. (Eds.). *Proceedings of the Fourth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'19). Advances in Intelligent Systems and Computing*. Springer; 2020. V. I. P. 131–141. https://doi.org/10.1007/978-3-030-50097-9_14
2. Baranauskas R., Janaviciute A., Jasinevicius R., Jukavicius V. On Multi-Agent Systems Intellectics. *Inf. Technol. Control*. 2015;44(1):112–121. <https://doi.org/10.5755/j01.itc.44.1.8768>
3. Darweesh S., Shehata H. Performance Evaluation of a Multi-Agent System using Fuzzy Model. *2018 First International Workshop on Deep and Representation Learning (IWDRL)*. 2018. P. 7–12. <https://doi.org/10.1109/IWDRL.2018.8358208>
4. Russel S., Norvig P. *Iskusstvennyi intellekt: sovremenniy podkhod*. T. 2. Znaniya i rassuzhdeniya v usloviyakh neopredelennosti (*Artificial Intelligence: A Modern Approach*. V. 2. Knowledge and Reasoning under Uncertainty): transl. from Engl. St. Petersburg: Dialektika; 2021. 480 p. (in Russ.).
5. Russel S., Norvig P. *Iskusstvennyi intellekt: sovremenniy podkhod*. T. 3. Obuchenie, vospriyatie i deistvie (*Artificial Intelligence: A Modern Approach*. V. 3. Learning, Perception, and Action): transl. from Engl. St. Petersburg: Dialektika; 2022. 640 p. (in Russ.).
6. Sutton R.S., Barto E.J. *Obuchenie s podkrepleniem (Reinforcement Learning)*: transl. from Engl. Moscow: DMK Press; 2020. 552 p. (in Russ.).
7. Graesser L., Keng W.L. *Glubokoe obuchenie s podkrepleniem: teoriya i praktika na yazyke Python (Reinforcement Learning: Theory and Practice in Python)*: transl. from Engl. St. Petersburg: Piter; 2022. 416 p. (in Russ.). [Graesser L., Keng W.L. *Foundations of Deep Reinforcement Learning: Theory and Practice in Python*. Addison-Wesley Professional; 2020. 416 p.]
8. Pumperla M., Ferguson K. *Glubokoe obuchenie i igra v go (Deep Learning and the Game of Go)*: transl. from Engl. Moscow: DMK Press; 2020. 372 p. (in Russ.).
9. Red'ko V.G. *Evolyutsiya, neironnye seti, intellekt: Modeli i kontseptsii evolyutsionnoi kibernetiki (Evolution, Neural Networks, Intelligence: Models and Concepts of Evolutionary Cybernetics)*. Moscow: LIBROKOM; 2013. 224 p. (in Russ.).
10. Winder P. *Reinforcement Learning. Industrial Applications of Intelligent Agents*. O'Reilly Media, Inc.; 2021. 382 p.

13. Зайцев Е.И. Методология представления и обработки знаний в распределенных интеллектуальных информационных системах. *Автоматизация и современные технологии*. 2008;1:29–34.
14. Batouma N., Sourrouille J.-L. Dynamic adaption of resource aware distributed applications. *Int. J. Grid Distrib. Comput.* 2011;4(2):25–42. URL: http://article.nadiapub.com/IJGDC/vol4_no2/3.pdf
15. Nurmatova E.V., Gusev V.V., Kotliar V.V. Analysis of the features of the optimal logical structure of distributed databases. In: *GRID Workshop Proceedings – GRID 2018 Selected Papers of the 8th International Conference “Distributed Computing and Grid-technologies in Science and Education.”* Dubna: 2018. V. 2267. P. 579–584. URL: <https://ceur-ws.org/Vol-2267/579-584-paper-111.pdf>
11. Houhamdi Z., Athamena B., Abuzaineddin R., Muhairat M. A Multi-Agent System for Course Timetable Generation. *TEM Journal*. 2019;8(1):211–221. <https://doi.org/10.18421/TEM81-30>
12. Aly S., Badoor H. Performance Evaluation of a Multi-Agent System using Fuzzy Model. *2018 First International Workshop on Deep and Representation Learning (IWDRL)*. 2018. P. 175–189. <https://doi.org/10.1109/IWDRL.2018.8358208>
13. Zaytsev E.I. Method of date representation and processing in the distributed intelligence information systems. *Avtomatizatsiya i sovremennye tekhnologii = Automation. Modern Technologies*. 2008;1:29–34 (in Russ.).
14. Batouma N., Sourrouille J.-L. Dynamic adaption of resource aware distributed applications. *Int. J. Grid Distrib. Comput.* 2011;4(2):25–42. Available from URL: http://article.nadiapub.com/IJGDC/vol4_no2/3.pdf
15. Nurmatova E.V., Gusev V.V., Kotliar V.V. Analysis of the features of the optimal logical structure of distributed databases. In: *GRID Workshop Proceedings - GRID 2018 Selected Papers of the 8th International Conference “Distributed Computing and Grid-technologies in Science and Education”*. Dubna; 2018. V. 2267. P. 579–584. Available from URL: <https://ceur-ws.org/Vol-2267/579-584-paper-111.pdf>

Об авторах

Зайцев Евгений Игоревич, к.т.н., доцент, кафедра «Аппаратное, программное и математическое обеспечение вычислительных систем» Института кибербезопасности и цифровых технологий ФГБОУ ВО «МИРЭА – Российский технологический университет» (107996, Россия, Москва, ул. Стромынка, д. 20). E-mail: zajcev@mirea.ru. Scopus Author ID 57218190023, ResearcherID ABA-4823-2020, SPIN-код РИНЦ 9662-7658, <https://orcid.org/0000-0002-1979-5611>

Нурматова Елена Вячеславовна, к.т.н., доцент, кафедра «Аппаратное, программное и математическое обеспечение вычислительных систем» Института кибербезопасности и цифровых технологий ФГБОУ ВО «МИРЭА – Российский технологический университет» (107996, Россия, Москва, ул. Стромынка, д. 20). E-mail: nurmatova@mirea.ru. Scopus Author ID 57205460003, ResearcherID GQI-3212-2022, SPIN-код РИНЦ 7036-3661, <https://orcid.org/0000-0001-8511-0978>

About the authors

Evgeniy I. Zaytsev, Cand. Sci. (Eng.), Associate Professor, Department of Hardware Software and Mathematical Support of Computing System, Institute for Cybersecurity and Digital Technologies, MIREA – Russian Technological University (20, Stromynka ul., Moscow, 107996 Russia). E-mail: zajcev@mirea.ru. Scopus Author ID 57218190023, ResearcherID ABA-4823-2020, RSCI SPIN-code 9662-7658, <https://orcid.org/0000-0002-1979-5611>

Elena V. Nurmatova, Cand. Sci. (Eng.), Associate Professor, Department of Hardware Software and Mathematical Support of Computing System, Institute for Cybersecurity and Digital Technologies, MIREA – Russian Technological University (20, Stromynka ul., Moscow, 107996 Russia). E-mail: nurmatova@mirea.ru. Scopus Author ID 57205460003, ResearcherID GQI-3212-2022, RSCI SPIN-code 7036-3661, <https://orcid.org/0000-0001-8511-0978>