

**Технологии информационного общества****ПРОГРАММНАЯ РЕАЛИЗАЦИЯ РЕДАКТОРА МУЛЬТИАГЕНТНЫХ  
НЕЙРОКОГНИТИВНЫХ АРХИТЕКТУР**

Статья рекомендована к публикации членом редакционного совета А. В. Богдановым 27.03.2025.

**Бжихатлов Кантемир Чамалович**

*Кандидат физико-математических наук*

*Кабардино-Балкарский научный центр российской академии наук, лаборатория «Нейрокогнитивные автономные интеллектуальные системы», заведующий лабораторией*

*Нальчик, Российская Федерация*

*haosit13@mail.ru*

**Пшенокова Инна Ауесовна**

*Кандидат физико-математических наук*

*Институт информатики и проблем регионального управления – филиал КБНЦ РАН, лаборатория «Интеллектуальные среды обитания», заведующая лабораторией*

*Нальчик, Российская Федерация*

*pshenokova\_inna@mail.ru*

**Мамбетов Идар Арсенович**

*Кабардино-Балкарский научный центр российской академии наук, Инжиниринговый центр, младший научный сотрудник*

*Нальчик, Российская Федерация*

*idar.mam12@mail.ru*

**Канкулов Султан Ахмедович**

*Кабардино-Балкарский научный центр российской академии наук, лаборатория «Нейрокогнитивные автономные интеллектуальные системы», младший научный сотрудник*

*Нальчик, Российская Федерация*

*skankulov@mail.ru*

**Аталиков Борис Анзорович**

*Кабардино-Балкарский научный центр российской академии наук, лаборатория «Нейрокогнитивные автономные интеллектуальные системы», младший научный сотрудник*

*Нальчик, Российская Федерация*

*atalikov10@gmail.com*

**Аннотация**

Исследование посвящено разработке алгоритмов и программного обеспечения для моделирования процессов рассуждения и принятия решений на основе мультиагентных нейрокогнитивных архитектур. В основе подхода лежит абстракция взаимодействия нейронов мозга, где каждый нейрон рассматривается как рациональный агент, максимизирующий свою целевую функцию. Программное обеспечение реализовано по клиент-серверной архитектуре: клиент – редактор для визуализации и редактирования агентов, сервер – ядро, моделирующее взаимодействие агентов и управляющее их поведением. Архитектура поддерживает 3D-визуализацию и сетевой обмен сообщениями. Дальнейшая работа направлена на улучшение визуализации и оптимизацию отображения большого числа агентов.

---

© Бжихатлов К. Ч., Пшенокова И. А., Мамбетов И. А., Канкулов С. А., Аталиков Б. А., 2026

Производство и хостинг журнала «Информационное общество» осуществляется Институтом развития информационного общества.

Данная статья распространяется на условиях международной лицензии Creative Commons «Атрибуция — Некоммерческое использование — На тех же условиях» Всемирная 4.0 (Creative Commons Attribution – NonCommercial - ShareAlike 4.0 International; CC BY-NC-SA 4.0). См. <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.ru>

[https://doi.org/10.52605/16059921\\_2026\\_01\\_108](https://doi.org/10.52605/16059921_2026_01_108)

## Ключевые слова

*интеллектуальные системы; нейрокогнитивные архитектуры; программное обеспечение; мультиагентное моделирование; визуализация*

## Введение

Одним из ключевых направлений современных исследований в области искусственного интеллекта является разработка мультиагентных систем, позволяющих моделировать целенаправленное поведение. Востребованность подобных подходов объясняется масштабируемостью, надежностью, гибкостью и эффективностью мультиагентных систем [1]. Под мультиагентной системой понимается вычислительная структура, состоящая из множества автономных агентов, взаимодействующих в среде, которая может быть динамичной, непредсказуемой и открытой для достижения общих и собственных целей. При этом, чтобы гарантировать достижение общей цели проектирования мультиагентной системы, необходимо регулировать и координировать поведение отдельных агентов и их взаимодействие [2, 3]. Поэтому для моделирования мультиагентных систем требуется разработка специального программного обеспечения, которое способно обеспечить реализацию отдельных агентов и механизмов контроля и регулирования их поведения [4, 5]. Кроме того, необходимы инструменты для тестирования и верификации программ, реализующих мультиагентные системы [6].

Широкое внедрение мультиагентных систем приводит к активному развитию соответствующего программного обеспечения [7, 8]. Несмотря на очевидное сходство подобных программ, они зачастую предоставляют программные конструкции для реализации разных наборов концепций и абстракций агента [9]. Например, в работе [10] приведено подробное описание применения платформы, использующей три отдельные системы для моделирования разных уровней абстракции (Jason – уровень отдельного агента, SArtAgO – уровень среды, и Moise – уровень организации взаимодействия). Подобное разбиение на отдельные уровни абстракции позволяет программировать поведение агентов отдельно от разработки моделей среды и взаимодействия агентов со средой. Свою программную реализацию модели целенаправленного агента представили и авторы работы [11], в которой описана реализация модели поведения агентов на основе подхода «вера-желание-намерение» («belief-desire-intention», BDI). Стоит отметить, что представленная авторами среда моделирования написана в виде открытого программного обеспечения и использует достаточно распространенные технологии программирования. В статье представлен двухсторонний интерфейс взаимодействия моделей агентов с распространённым набором программных решений, объединённых «операционной системой для робототехнических устройств» ROS. При этом стоит отметить, что на данный момент существует ряд специализированных языков программирования (JaCaMo, Jason, GOAL, G WENDOLEN, 2APL и др.) для разработки моделей поведения агентов [12], используемых в зависимости от используемого формализма и условий применения результатов моделирования. Дальнейшее использование агентов, на основе подхода BDI в задачах моделирования управления автономными роботами в киберфизических описано в статьях [13, 14]. В [14] представлена архитектура, которая охватывает гетерогенные компоненты системы и соответствующие уровни проектирования, включая кибер- и физические аспекты, и интегрирует их с высокоуровневым механизмом рассуждений для киберфизических систем. В [15] представлена программная реализация универсального реактивного агента на языке GOAL с использованием сценария Agents Assemble. В такой реализации каждый агент основан на одном и том же наборе логических правил, однако это может привести к заикливанию агентов, при котором у них нет четких действий для выхода из тупика. Одним из возможных выходов проблемы может стать назначение агентам различных ролей или объединение их в небольшие команды, которые работают вместе.

Следует отметить, что мультиагентные модели имеют крайне широкий спектр применения. Например, в работе [16] представлены мультиагентные модели управления группировкой космических аппаратов на основе концепции мультиагентной сети потребностей и возможностей. Суть разработанного метода виртуального рынка состоит в поиске лучшего решения для каждой заявки, которое может нарушать ограничения по совместному использованию ресурсов, и последовательного итерационного приближения к нему путем разбора конфликтов и взаимных уступок агентов с выплатой компенсаций. Подобный подход позволит обеспечить эффективное управление группой агентов в режиме разделения заказов и ресурсов. Кроме того, мультиагентные алгоритмы могут использоваться для моделирования процесса принятия решений [17-19].

В данной работе представлено программное обеспечение для взаимодействия с имитационной моделью интеллектуальных систем на основе мультиагентных нейрокогнитивных архитектур, описанных в [17]. Этот подход к проектированию интеллектуальных систем принятия решений состоит в синтезе когнитивных архитектур и мультиагентных систем. Причем, в качестве когнитивной архитектуры используется архитектура, в которой составляющие ее узлы представляют собой мультиагентные системы. Данный подход опирается на вычислительную абстракцию процессов мультиагентного обмена информацией между нейронами головного мозга, в которой отдельные нейроны рассматриваются как рациональные программные агенты, выполняющие кооперативное взаимодействие друг с другом с целью максимизации своих локальных целевых функций [18]. В силу того, что в когнитивной архитектуре, основанной на использовании подобной вычислительной абстракции, когнитивные состояния детерминированы мультиагентным взаимодействием нейроподобных элементов в составе функциональных узлов, данная архитектура получила название мультиагентной нейрокогнитивной архитектуры [19].

Мультиагентная нейрокогнитивная архитектура определяется как рекурсивная когнитивная архитектура (допускающая вложенность агентов и функциональных систем друг в друга), функциональные узлы которой состоят из программных агентов-нейронов (*агнейронов*) разной степени сложности и объединены в составе т.н. инварианта организационной структуры принятия решений на основе нейрокогнитивной архитектуры (инвариант когнитивной архитектуры) [20]. Этот инвариант включает в себя последовательность обязательных операций: *распознавания состояний, идентификации и оценки проблемных ситуаций, синтеза целевых состояний и путей их достижения*. Каждая из этих операций выполняется на основе *мультиагентного алгоритма*, основанного на обмене сообщениями между агнейронами различных типов, расположенными в соответствующих функциональных узлах нейрокогнитивной архитектуры. Каждый агнейрон обладает базой знаний, согласно которой функционирует и заключает *мультиагентные контракты*. *Мультиагентные знания* состоят из набора правил, которые представляют собой продукции, условная часть которых определяет начальную и конечную ситуацию, а ядро – действие, которое переводит агента из начальной ситуации в конечную. Если в системе нет агнейрона, отвечающего за обработку поступившего нового сообщения или события, в архитектуре они порождаются «по требованию» специальными агнейронами-фабриками. Фабрики порождают агнейронов определенного типа с первичным набором мультиагентных знаний (*геном агнейрона*), который в процессе обучения и взаимодействия с другими агнейронами модифицируется. Так как система рекурсивная, то каждый агнейрон содержит такие же функциональные узлы, состоящие из агентов – акторов. Преимуществом такой когнитивной архитектуры является ее полное архитектурное соответствие головному мозгу человека. Что же касается функционального соответствия, то в работе [20] мультиагентная нейрокогнитивная архитектура была применена для исследования возможностей его обеспечения в задачах интеллектуального управления автономными агентами, в [21, 22] интерпретации естественно-языковых высказываний, в [23] синтеза децентрализованного управления кооперативным поведением гетерогенных коллективов.

Объектом исследования являются алгоритмы и программное обеспечение моделирования процесса рассуждения и принятия решений на основе мультиагентных нейрокогнитивных архитектур.

Цель работы – разработка алгоритмов работы и реализация программного обеспечения для создания, редактирования и визуализации процесса рассуждения и принятия решений на основе мультиагентных нейрокогнитивных архитектур.

## **1 Архитектура взаимодействия редактора и программы моделирования мультиагентных нейрокогнитивных архитектур**

Учитывая высокие требования к вычислительной мощности ЭВМ, обеспечивающего моделирование мультиагентных нейрокогнитивных архитектур, наиболее эффективным подходом к организации взаимодействия программы для моделирования с пользователем является реализация клиент-серверной архитектуры, где в качестве клиентов могут выступать программы для просмотра и редактирования архитектуры, а также программно-аппаратные комплексы, использующие результаты моделирования. Схема работы редактора мультиагентных нейрокогнитивных архитектур, в этом случае, показана на рисунке 1. Для обеспечения функционала редактор в первую очередь обеспечивает сетевой обмен сообщениями с программой

для моделирования (ядром мультиагентной нейрокогнитивной архитектуры). Программа моделирования представляет собой сервер, на котором ведется моделирование процессов взаимодействия отдельных агентов. Ядро мультиагентной нейрокогнитивной архитектуры отвечает за работу с базой данных, содержащей список моделируемых агентов, за моделирование процесса обмена сообщениями и энергией, а также за обработку сетевых запросов от клиентов. Программа написана на языке программирования C++ с использованием библиотек Qt и базы данных MongoDB.

Полученные редактором с ядра данные позволяют составить описание состояния моделируемой архитектуры (для чего используется структуры данных, схожих с организацией структуры интеллектуального агента). Это, в свою очередь, позволяет построить визуализацию, использующую трехмерное представление всех агентов и связей между ними. Кроме визуализации модели, в редактор добавлен функционал для обмена сообщениями с интеллектуальным агентом (*чат*) и система отслеживания состояния агентов в архитектуре (*дебагер*). Главным инструментом для редактирования поведения отдельных агентов и их групп выступают модули редактирования правил и геномов.

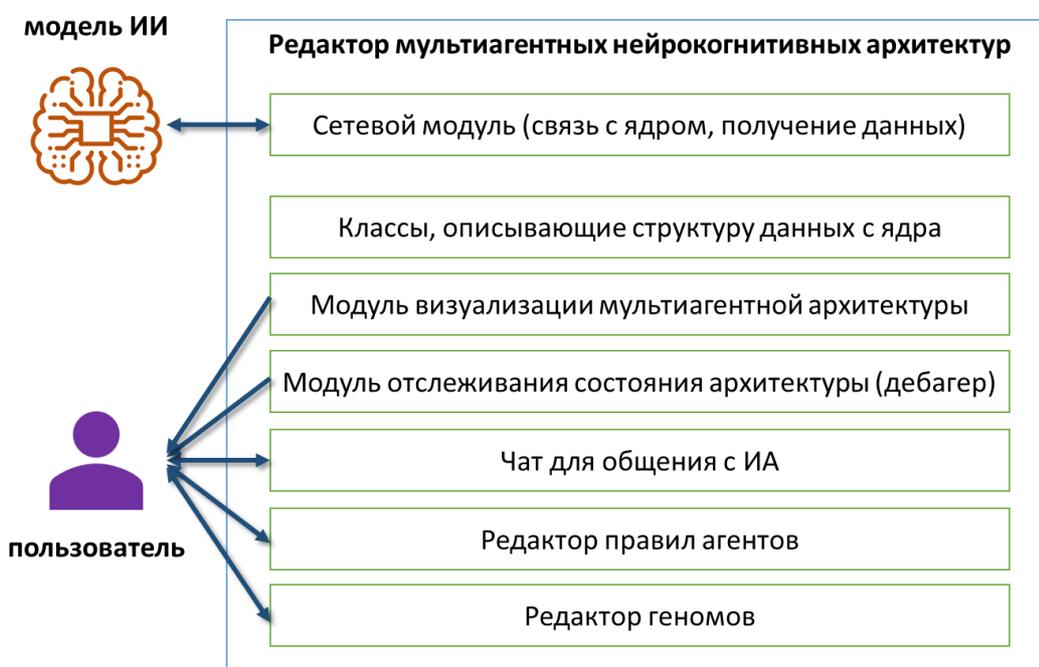


Рис. 1. Схема работы редактора мультиагентных нейрокогнитивных архитектур

При этом стоит учесть, что редактор должен обеспечивать отправку запросов на программу моделирования. В качестве ответа на подобные запросы могут выступать информация о состоянии интеллектуального агента (полная модель мультиагентной нейрокогнитивной архитектуры) и список вновь сгенерированных сообщений, что позволит редактору получать все необходимые данные для визуализации по запросу. Кроме того, для обеспечения оперативного отображения важных событий, ядро без запроса со стороны редактора может отправить информацию об изменении состояния определенных агентов и дополнительные данные для отладки. Со стороны ядра, в свою очередь, могут быть отправлены запросы с данными для редактирования агентов в модели (например, для изменения базы знаний отдельного нейрона), а также команды для настройки и управления процессом моделирования. Схема обмена данными между редактором и программой для моделирования показана на рисунке 2.

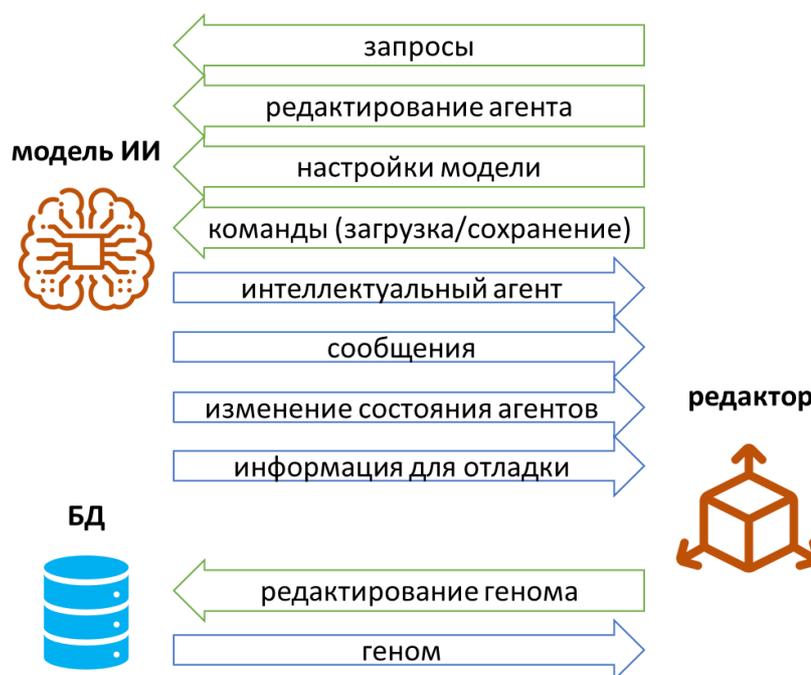


Рис. 2. Схема обмена данными между редактором и программой для моделирования мультиагентных нейрокогнитивных архитектур

Стоит отметить, что кроме работы с программой для моделирования мультиагентной архитектуры редактор имеет возможность работы с базой данных, в которой хранится описание «генома» агентов архитектуры. Такой подход позволит обеспечить работоспособность редактора даже при отключенном процессе моделирования какой-либо мультиагентной архитектуры.

## 2 Программная реализация редактора мультиагентных нейрокогнитивных архитектур

Поскольку отображение мультиагентной архитектуры предполагает использование трехмерной визуализации, для разработки редактора использовалось трехмерный движок, в частности, текущая версия собрана на базе бесплатного игрового движка Unity 3D, а программа написана на языке программирования C#. Обмен сообщениями реализован за счет использования сетевого протокола WebSocket (на основе TCP/IP соединения). После запуска программы и инициализации сцены (виртуального трехмерного пространства для отображения всех необходимых элементов) редактор запрашивает содержимое интеллектуального агента у доступных по сети программ моделирования мультиагентных архитектур. Полученные данные (в виде json файла) становятся основой для отображения трехмерной модели интеллектуального агента. Структура ответа с данными о интеллектуальном агенте показана на рисунке 3.

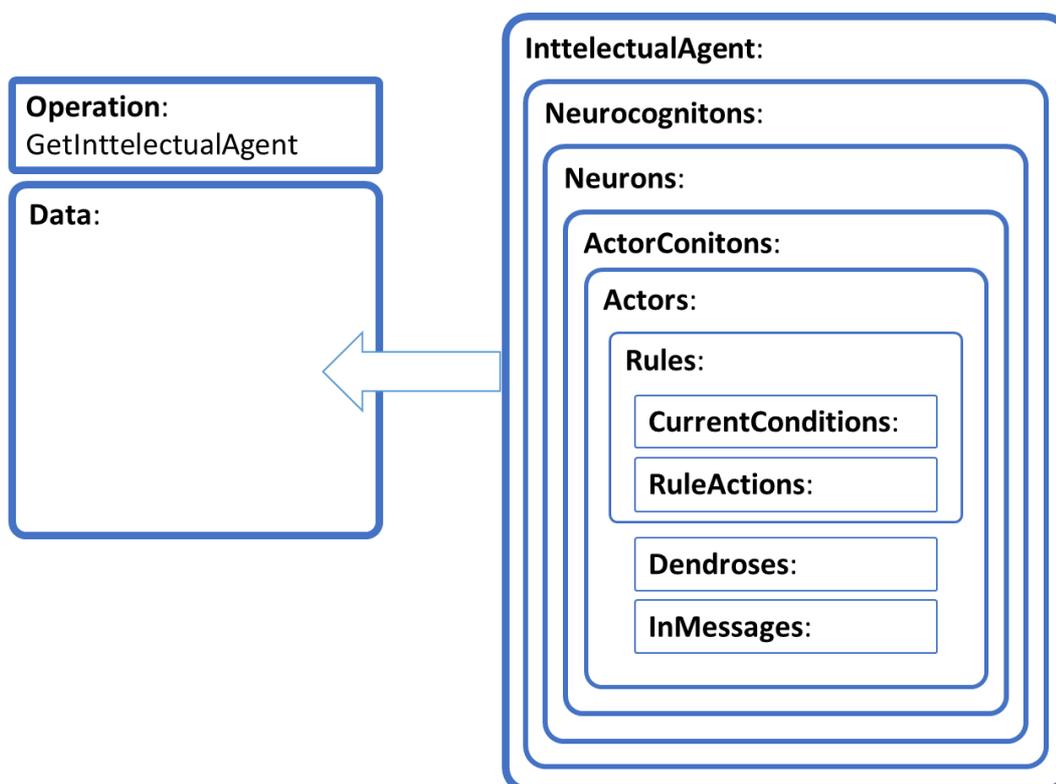


Рис. 3. Структура данных с описанием интеллектуального агента

Каждое сообщение между редактором и ядром имеет базовую структуру, состоящую из полей Operation и Data. Первое поле содержит тип запроса или ответа. В случае получения данных о содержимом интеллектуального агента – это запись «GetIntellectualAgent». Структура поля Data подстраивается для каждой команды. В частности, при запросе интеллектуального агента в этом поле содержится его описание, состоящее из списка нейрокогнитонов, каждый из которых включает в себя список нейронов и т.д.

Для уменьшения объема передаваемых данных, кроме структуры с интеллектуальным агентом, ядро может отправить структуру, содержащую новые сгенерированные агентами сообщения (рисунок 4). В этом случае в поле Operation содержится команда «NewMessage», а в поле Data список новых сообщений, каждое из которых включает данные об отправителе, получателе, типе и содержимом сообщения и времени его жизни.

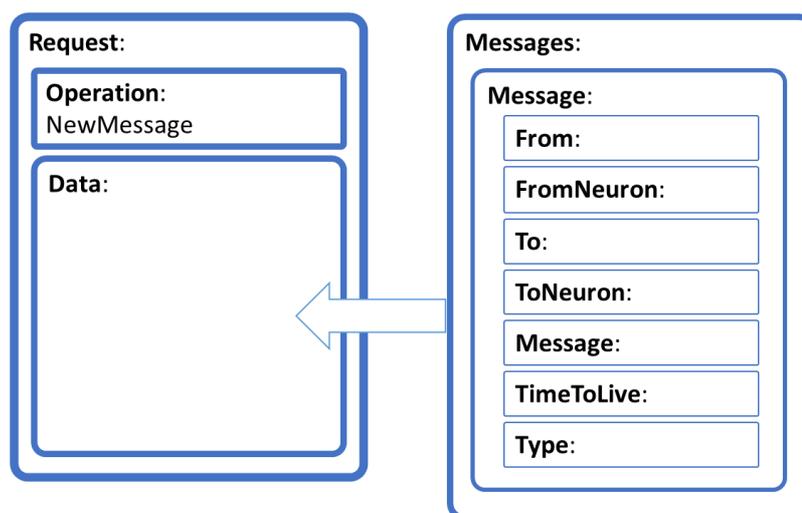


Рис. 4. Структура данных с описанием новых сообщений

Эти данные позволяют обеспечить отображение процессов, происходящих в имитационной модели мультиагентной нейрокогнитивной системы принятия решений в режиме реального

времени без значительного повышения нагрузки на сетевую инфраструктуру. Сам процесс отображения предназначен для максимально наглядной визуализации моделируемой архитектуры и процессов, происходящих в ней. Для этого на сцене показаны все элементы интеллектуального агента, связи между ними и сгенерированные агентами сообщения. При этом движок Unity поддерживает перемещение камеры пользователя по всей сцене, что позволяет подобрать наиболее удобные для работы ракурсы.

При визуализации агента каждый нейрокогнитон (тип нейронов) располагается на отдельном слое. Для визуализации слоев отображаются две полупрозрачные плоскости, отделяющие все нейроны нейрокогнитона от остальных моделей. Соответственно, на каждый слой расставляются модели фабрики нейронов, обобщенного нейрона и всех нейронов нейрокогнитона (с учетом заполнения слоя). Кроме того, для каждого нейрона отрисовываются все исходящие сообщения (в виде набора объектов под нейроном) и его контракты с другими нейронами (в виде соединений между моделями нейронов). Блок-схема алгоритма визуализации показана на рисунке 5.



Рис. 5. Алгоритм отображения нейронов на сцене

Стоит отметить, что программа поддерживает рекурсивность архитектуры и позволяет просмотреть содержимое каждого нейрона. Алгоритм отображения акторов схож с алгоритмом для нейронов: происходит перебор всех акторкогнитонов определенного нейрона и акторов в них.

### 3 Интерфейс редактора мультиагентных нейрокогнитивных архитектур

Кроме визуализации модели редактор предназначен и для управления процессом редактирования, а также изменения правил в базах знаний агентов. При этом, учитывая сложность моделируемых процессов, возможность редактирования элементов также реализована как в виде отдельных окон, так и в режиме взаимодействия с трехмерными объектами. Внешний вид главного окна программы показан на рисунке 6. После запуска, программа просит выбрать моделируемого интеллектуального агента из списка и некоторые параметры процесса моделирования (например, скорость работы модели). Основную часть окна программы составляет трехмерная сцена, на которой видны нейроны текущего интеллектуального агента. В верхней части основного окна расположено главное меню, откуда есть доступ к функционалу программы, в частности к запуску и остановке моделирования, настройкам процесса моделирования, к функциям создания новых агентов в модели, настройками редактора и функционалу редактирования агентов. Кроме этого, на окне отображается несколько отдельных модулей, предназначенных для взаимодействия с моделью.

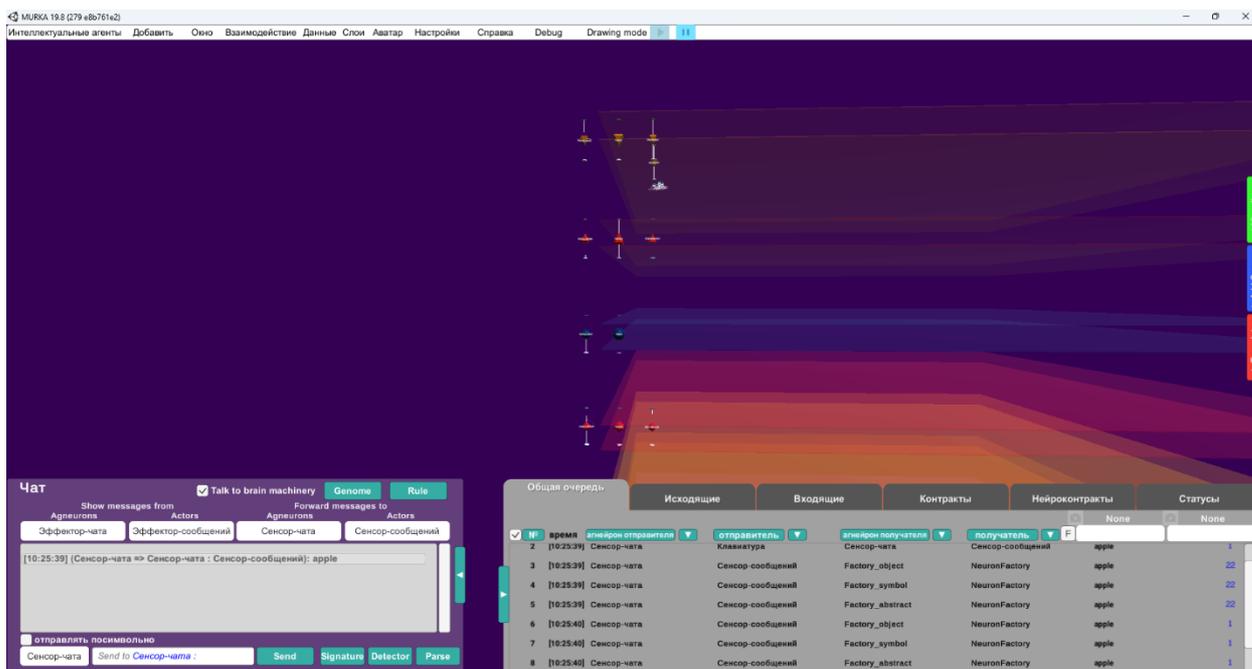


Рис. 6. Скриншот основного редактора мультиагентных нейрокогнитивных архитектур

Каждый агент представлен в виде трехмерного объекта на сцене, который позволяет как отслеживать его состояние, так и взаимодействовать с ним. Например, модель нейрона, показанная на рисунке 7, представляет собой трехмерный объект, состоящий из *сомы* (основного тела, внешний вид которого определяется нейрокогнитивной моделью), аксона и дендрита. Рядом отображаются модели сообщений, размер которых коррелирует с временем жизни, оставшимся у сообщения. С аксоном и дендритом отображаются связи (мультиагентные контракты) между нейронами. При этом, нажатие на сому активирует вызов меню редактирования агента, позволяющее изменить его свойства или добавить новое правило в базу знаний.



Рис. 7. Внешний вид модели актора с его основными узлами

Для обмена сообщениями между пользователем и мультиагентной архитектурой используется соответствующий модуль чата, интерфейс которого показан на рисунке 8. Чат позволяет писать текстовые сообщения нейронам, отвечающим за обмен данными с внешней средой (по умолчанию это «Сенсор-сообщений» и «Эффектор-сообщений»). А при необходимости можно генерировать сообщения для конкретного актора в архитектуре. Здесь же находятся кнопки вызова редакторов правил для акторов и генома.

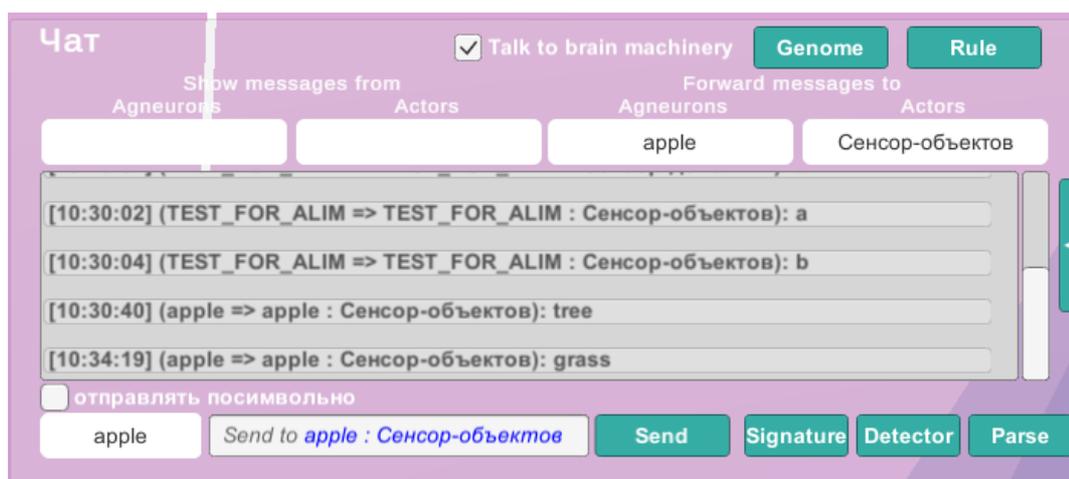


Рис. 8. Внешний вид модуля текстового взаимодействия с архитектурой

Для отладки разрабатываемых архитектур используется модуль дебагера (рисунок 9). В нем, в отличие от чата, отображаются все сообщения от всех агентов в архитектуре. Сюда же выводятся служебные сообщения от ядра, связанные с созданием новых агентов, заключение контрактов между ними, гибелью агентов и т.д. Для удобства в данном модуле реализована фильтрация сообщений по нескольким полям: отправитель, получатель, содержимое и тип сообщения (кнопки в верхней части списка сообщений). Кроме отслеживания общего потока сообщений в дебагере есть

возможность просмотра сообщений и контрактов для определенного актора, указанного в двух списках в правой части дебагера.

№	время	агнейрон отправителя	отправитель	агнейрон получателя	получатель	контракт	статус
32	[10:30:39]	TEST_FOR ALIM	Сенсор-действия	TEST_FOR ALIM	Factory-action	a	186
33	[10:30:41]	apple	Клавиатура	apple	Сенсор-объектов	tree	1
34	[10:30:41]	apple	Клавиатура	apple	Сенсор-объектов	tree	1
35	[10:30:41]	apple	Сенсор-объектов	apple	object*	tree	1
36	[10:30:41]	apple	Сенсор-объектов	apple	Factory-object	tree	97
37	[10:30:54]	TEST_FOR ALIM	Сенсор-объектов	TEST_FOR ALIM	Factory-object	a	97
38	[10:30:56]	TEST_FOR ALIM	Сенсор-объектов	TEST_FOR ALIM	Factory-object	b	94

Рис. 9. Внешний вид модуля отладки работы архитектуры

Редактирование базы знаний агентов осуществляется в окне редактора правил, показанного на рисунке 10. В нем, после указания нейрона и актора, можно просмотреть весь список правил агента, представленный в виде наборов условных и результирующих клауз. Пользователь может внести изменения в любую часть правила, например, изменить адресата, содержимое сообщения, время жизни сгенерированного сообщения, добавить новую клаузу, добавить или удалить правило. При нажатии кнопки сохранения – все изменения записываются в моделируемую архитектуру и изменяют дальнейшее поведение выбранного актора.

apple + - [copy] [paste] [undo] [redo] obj\_1 + - [copy] [paste] [undo] [redo]

Enable Правило 2 + - [copy] [paste] [undo] [redo] 0

Отправитель Если

NO [input] @ [input] @ Default DENDROS\_ACTIVAT Min Max &

Получатель Тогда

& [input] @ action\* Default Myname 20 [copy]

& [input] @ action\* Default Продам 20 [copy]

& [input] @ StartMySell StartMySellerTest;50 5000 [copy]

+ - [input] @ Дисплей Default Myname 20 [copy]

Save Show agent Update

Visualizer Add Sensors

Рис. 10. Внешний вид редактора правил агентов

## Заключение

В рамках разработки программного обеспечения для моделирования процесса рассуждения и принятия решений был разработан редактор мультиагентных нейрокогнитивных архитектур, предназначенный для визуализации процесса моделирования, взаимодействия с моделью и редактирования отдельных агентов в ней. Для организации взаимодействия программы с пользователем была реализована клиент-серверная архитектура. В качестве клиентов выступает программа для просмотра и редактирования интеллектуальных агентов (редактор мультиагентных нейрокогнитивных архитектур). Разработанный редактор обеспечивает сетевой обмен сообщениями с программой, обеспечивающей реализацию интеллектуальных агентов и механизмов контроля и регулирования их поведения (ядро мультиагентных нейрокогнитивных архитектур). В работе представлены программная реализация и интерфейс редактора мультиагентных нейрокогнитивных архитектур.

Дальнейшая доработка программного обеспечения направлена на улучшение трехмерной визуализации и оптимизацию отображения агентов, что позволит обеспечить стабильную работу

редактора в случае моделирования мультиагентных архитектур, включающих большое количество агентов.

## Литература

1. Dorri A., Kanhere S. S., Jurdak R. Multi-agent systems: A survey // *Ieee Access*, 2018, vol. 6, pp. 28573-28593. DOI: <https://doi.org/10.1109/ACCESS.2018.2831228>.
2. Юлейси Г. П., Холод И. И. Взаимодействие в многоагентных системах интеллектуального анализа данных // *Известия СПбГЭТУ ЛЭТИ*. 2020. № 3. С. 18-23.
3. Зайцев Е.И., Нурматова Е.В. О подходе к управлению знаниями и разработке мультиагентной системы представления и обработки знаний. *Russian Technological Journal*. 2023. 11(4). С.16-25. DOI: <https://doi.org/10.32362/2500-316X-2023-11-4-16-25>.
4. Бурлуцкий В. В., Керамов Н. Д., Балуев В. А., Изерт М. И., Якимчук А. В. Разработка мультиагентной интеллектуальной системы для решения задач классификации и ранжирования материалов в сети интернет // *Вестник югорского государственного университета*. 2020. Т. 3 (58). С. 47-52.
5. Латышев А.К., Панов А.И. Методы внутренней мотивации в задачах обучения с подкреплением на основе модели // *Искусственный интеллект и принятие решений*. 2023. № 3. С. 84-97.
6. Sujil A., Verma J., Kumar R. Multi agent system: concepts, platforms and applications in power systems // *Artificial Intelligence Review*, 2018, vol. 49, pp. 153-182.
7. Bordini R. H. et al. Agent programming in the cognitive era // *Autonomous Agents and Multi-Agent Systems*, 2020, vol. 34, pp. 1-31.
8. Abar S. et al. Agent Based Modelling and Simulation tools: A review of the state-of-art software // *Computer Science Review*, 2017, vol. 24, pp. 13-33.
9. Бежитская Е. А., Бежитский С. С., Казанцева П. И. Обзор и сравнение программных средств для реализации многоагентных систем // *Решетневские чтения*. 2018. Т. 2. С. 102-103.
10. Boissier O. et al. Multi-agent oriented programming: programming multi-agent systems using JaCaMo. *Mit Press*, 2020, 235 p.
11. Cardoso R. C. et al. An interface for programming verifiable autonomous agents in ROS // *European Conference on Multi-Agent Systems*. Cham: Springer International Publishing, 2020, pp. 191-205.
12. Cardoso R. C., Ferrando A. A review of agent-based programming for multi-agent systems // *Computers*, 2021, vol. 10, № 2, pp. 16-26.
13. Cardoso, R.C., Ferrando, A., Collette, J., Dennis, L.A., Fisher, M. Towards Forward Responsibility in BDI Agents. In: Ciorrea, A., Dastani, M., Luo, J. (eds) *Engineering Multi-Agent Systems*. EMAS 2023. *Lecture Notes in Computer Science*, 2023, vol 14378. Springer, Cham. DOI: [https://doi.org/10.1007/978-3-031-48539-8\\_1](https://doi.org/10.1007/978-3-031-48539-8_1).
14. Karaduman B., Tezel B. T., Challenger M. Rational software agents with the BDI reasoning model for Cyber-Physical Systems // *Engineering Applications of Artificial Intelligence*. 2023, vol. 123, pp. 106478.
15. Jensen A. B., Villadsen J. GOAL-DTU: development of distributed intelligence for the multi-agent programming contest // *The Multi-Agent Programming Contest 2019: Agents Assemble-Block by Block to Victory 14*. Springer International Publishing, 2020, pp. 79-105.
16. Галузин В.А., Галицкая А.В., Скобелев П.О. Мультиагентные технологии для управления орбитальной группировкой малых космических аппаратов // В сборнике: XVI Всероссийская мультиконференция по проблемам управления (МКПУ-2023). Волгоград, 2023. С. 136-138.
17. Nagoev Z., Nagoeva O., Gurtueva I. Multi-agent neurocognitive models of semantics of spatial localization of events. *Cognitive Systems Research*, 2020, vol.59, pp. 91-102.
18. Nagoev Z., Pshenokova I., Anchekov M. Model of the reasoning process in a multiagent cognitive system. *Procedia Computer Science*, 2020, vol. 169, pp. 615-619.
19. Nagoev Z., Pshenokova I., Nagoeva O., Sundukov Z.: Learning algorithm for an intelligent decision-making system based on multi-agent neurocognitive architectures. *Cognitive Systems Research*, 2021, vol. 66, pp. 82-88.

20. Pshenokova, I., Bzhikhatlov, K., Kankulov, S., Apshev, A., Atalikov, B. Simulation Model of the Neurocognitive System Controlling an Intellectual Agent Displaying Exploratory Behavior in the Real World. *Studies in Computational Intelligence*, 2024, vol. 1130. Springer, Cham. DOI: [https://doi.org/10.1007/978-3-031-50381-8\\_76](https://doi.org/10.1007/978-3-031-50381-8_76).
21. Zalikhan Nagoev, Olga Nagoeva, Murat Anchokov, Kantemir Bzhikhatlov, Sultan Kankulov, Ahmed Enes The symbol grounding problem in the system of general artificial intelligence based on multi-agent neurocognitive architecture. *Cognitive Systems Research*, 2023, vol. 79, pp. 71-84, DOI: <https://doi.org/10.1016/j.cogsys.2023.01.002>.
22. Нагоев З. В., Нагоева О. В. Обоснование символов и мультиагентные нейрокогнитивные модели семантики естественного языка. Нальчик: Издательство КБНЦ РАН, 2022 г., 150 с.
23. Pshenokova I., Bzhikhatlov K., Nagoeva O., Mambetov I., Unagasov A. Autonomous Robot Navigation System as Part of a Human-machine Team Based on Self-organization of Distributed Neurocognitive Architectures // In: Ronzhin, A., Sadigov, A., Meshcheryakov, R. (eds) *Interactive Collaborative Robotics. ICR 2023. Lecture Notes in Computer Science*, 2023, vol. 14214. Springer, Cham. DOI: [https://doi.org/10.1007/978-3-031-43111-1\\_6](https://doi.org/10.1007/978-3-031-43111-1_6).

# DEVELOPMENT OF ALGORITHMS AND SOFTWARE IMPLEMENTATION OF THE EDITOR OF MULTI-GROUP NEUROCOGNITIVE ARCHITECTURES

## **Bzhikhatlov, Kantemir Chamalovich**

*Candidate of physical and mathematical sciences*

*Kabardino-Balkarian Research Center of the Russian Academy of Sciences, laboratory "Neurocognitive autonomous intelligent systems", head of the laboratory*

*Nalchik, Russian Federation*

*haosit13@mail.ru*

## **Pshenokova, Inna Auesovna**

*Candidate of physical and mathematical sciences*

*Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences, Institute of computer science and problems of regional management, laboratory "Smart living environments", head of the laboratory*

*Nalchik, Russian Federation*

*pshenokova\_inna@mail.ru*

## **Mambetov, Idar Arsenovich**

*Kabardino-Balkarian Research Center of the Russian Academy of Sciences, Engineering center, junior researcher*

*Nalchik, Russian Federation*

*idar.mam12@gmail.com*

## **Kankulov, Sultan Akhmedovich**

*Kabardino-Balkarian Research Center of the Russian Academy of Sciences, laboratory "Neurocognitive autonomous intelligent systems", junior researcher*

*Nalchik, Russian Federation*

*skankulov@mail.ru*

## **Atalikov, Boris Anzorovich**

*Kabardino-Balkarian Research Center of the Russian Academy of Sciences, laboratory "Neurocognitive autonomous intelligent systems", junior researcher*

*Nalchik, Russian Federation*

*atalikov@gmail.com*

## **Abstract**

*The research focuses on developing algorithms and software for modeling reasoning and decision-making processes based on multi-agent neurocognitive architectures. The approach is based on an abstraction of brain neuron interactions, where each neuron is considered a rational agent maximizing its objective function. The software is implemented using a client-server architecture: the client is an editor for visualizing and editing agents, while the server is the core that simulates agent interactions and manages their behavior. The architecture supports 3D visualization and network message exchange. Future work aims to improve visualization and optimize the display of a large number of agents.*

## **Keywords**

*intelligent systems; neurocognitive architectures; software; multi-agent modeling; visualization*

## **References**

1. Dorri A., Kanhere S. S., Jurdak R. Multi-agent systems: A survey // Ieee Access, 2018, vol. 6, pp. 28573-28593. DOI: <https://doi.org/10.1109/ACCESS.2018.2831228>.
2. Yulejsi G. P., Holod I. I. Vzaimodejstvie v mnogoagentnyh sistemah intellektual'nogo analiza dannyh // Izvestiya SPbGETU LETI. 2020. №. 3. S. 18-23.
3. Zajcev E.I., Nurmatova E.V. O podhode k upravleniyu znaniyami i razrabotke mul'tiagentnoj sistemy predstavleniya i obrabotki znaniy. Russian Technological Journal. 2023. 11(4). S.16-25. DOI: <https://doi.org/10.32362/2500-316X-2023-11-4-16-25>.

4. Burluckij V. V., Keramov N. D., Baluev V. A., Izert M. I., Yakimchuk A. V. Razrabotka mul'tiagentnoj intellektual'noj sistemy dlya resheniya zadach klassifikacii i ranzhirovaniya materialov v seti internet // Vestnik yugorskogo gosudarstvennogo universiteta. 2020. T. 3 (58). S. 47-52.
5. Latyshev A.K., Panov A.I. Metody vnutrennej motivacii v zadachah obucheniya s podkrepleniem na osnove modeli // Iskusstvennyj intellekt i prinyatie reshenij. 2023. № 3. S. 84-97.
6. Sujil A., Verma J., Kumar R. Multi agent system: concepts, platforms and applications in power systems // Artificial Intelligence Review, 2018, vol. 49, pp. 153-182.
7. Bordini R. H. et al. Agent programming in the cognitive era // Autonomous Agents and Multi-Agent Systems, 2020, vol. 34, pp. 1-31.
8. Abar S. et al. Agent Based Modelling and Simulation tools: A review of the state-of-art software // Computer Science Review, 2017, vol. 24, pp. 13-33.
9. Bezhitskaya E. A., Bezhitskij S. S., Kazanceva P. I. Obzor i sravnenie programmnyh sredstv dlya realizacii mnogoagentnyh sistem // Reshetnevskie chteniya. 2018. T. 2. S. 102-103.
10. Boissier O. et al. Multi-agent oriented programming: programming multi-agent systems using JaCaMo. Mit Press, 2020, 235 p.
11. Cardoso R. C. et al. An interface for programming verifiable autonomous agents in ROS // European Conference on Multi-Agent Systems. Cham: Springer International Publishing, 2020, pp. 191-205.
12. Cardoso R. C., Ferrando A. A review of agent-based programming for multi-agent systems // Computers, 2021, vol. 10, № 2, pp. 16-26.
13. Cardoso, R.C., Ferrando, A., Collette, J., Dennis, L.A., Fisher, M. Towards Forward Responsibility in BDI Agents. In: Ciorrea, A., Dastani, M., Luo, J. (eds) Engineering Multi-Agent Systems. EMAS 2023. Lecture Notes in Computer Science, 2023, vol 14378. Springer, Cham. DOI: [https://doi.org/10.1007/978-3-031-48539-8\\_1](https://doi.org/10.1007/978-3-031-48539-8_1).
14. Karaduman B., Tezel B. T., Challenger M. Rational software agents with the BDI reasoning model for Cyber-Physical Systems // Engineering Applications of Artificial Intelligence. 2023, vol. 123, pp. 106478.
15. Jensen A. B., Villadsen J. GOAL-DTU: development of distributed intelligence for the multi-agent programming contest // The Multi-Agent Programming Contest 2019: Agents Assemble-Block by Block to Victory 14. Springer International Publishing, 2020, pp. 79-105.
16. Galuzin V.A., Galickaya A.V., Skobelev P.O. Mul'tiagentnye tekhnologii dlya upravleniya orbital'noj gruppirovkoj malyh kosmicheskikh apparatov // V sbornike: XVI Vserossiyskaya mul'tikonferenciya po problemam upravleniya (MKPU-2023). Volgograd, 2023. S. 136-138.
17. Nagoev Z., Nagoeva O., Gurtueva I. Multi-agent neurocognitive models of semantics of spatial localization of events. Cognitive Systems Research, 2020, vol.59, pp. 91-102.
18. Nagoev Z., Pshenokova I., Anchekov M. Model of the reasoning process in a multiagent cognitive system. Procedia Computer Science, 2020, vol. 169, pp. 615-619.
19. Nagoev Z., Pshenokova I., Nagoeva O., Sundukov Z.: Learning algorithm for an intelligent decision-making system based on multi-agent neurocognitive architectures. Cognitive Systems Research, 2021, vol. 66, pp. 82-88.
20. Pshenokova, I., Bzhikhatlov, K., Kankulov, S., Apshev, A., Atalikov, B. Simulation Model of the Neurocognitive System Controlling an Intellectual Agent Displaying Exploratory Behavior in the Real World. Studies in Computational Intelligence, 2024, vol. 1130. Springer, Cham. DOI: [https://doi.org/10.1007/978-3-031-50381-8\\_76](https://doi.org/10.1007/978-3-031-50381-8_76).
21. Zalimkhan Nagoev, Olga Nagoeva, Murat Anchokov, Kantemir Bzhikhatlov, Sultan Kankulov, Ahmed Enes The symbol grounding problem in the system of general artificial intelligence based on multi-agent neurocognitive architecture. Cognitive Systems Research, 2023, vol. 79, pp. 71-84, DOI: <https://doi.org/10.1016/j.cogsys.2023.01.002>.
22. Nagoev Z. V., Nagoeva O. V. Obosnovanie simvolov i mul'tiagentnye neirokognitivnye modeli semantiki estestvennogo yazyka. Nal'chik: Izdatel'stvo KBNC RAN, 2022 g., 150 s.
23. Pshenokova I., Bzhikhatlov K., Nagoeva O., Mambetov I., Unagasov A. Autonomous Robot Navigation System as Part of a Human-machine Team Based on Self-organization of Distributed Neurocognitive Architectures // In: Ronzhin, A., Sadigov, A., Meshcheryakov, R. (eds) Interactive Collaborative Robotics. ICR 2023. Lecture Notes in Computer Science, 2023, vol. 14214. Springer, Cham. DOI: [https://doi.org/10.1007/978-3-031-43111-1\\_6](https://doi.org/10.1007/978-3-031-43111-1_6).