УДК 004.89:005.53

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АРХИТЕКТУР СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ И ИХ ГИБРИДНОГО ОБЪЕДИНЕНИЯ

А. О. Костыренков, аспирант, ассистент каф. ИиППО ИИТ РТУ МИРЭА, Москва, Россия;
orcid.org/0009-0007-0294-694X, e-mail: kostyrenkov@mirea.ru
Э. Н. Мифтахов, д.ф.-м.н., профессор каф. ИиППО ИИТ РТУ МИРЭА, Москва, Россия;
orcid.org/0000-0002-0471-5949, e-mail: promif@mail.ru

Работа посвящена сравнительному анализу трёх архитектур систем поддержки принятия решений (СППР) для управления устройствами: правило-ориентированной, LLM-ориентированной и гибридной (LLM + правила). Особое внимание уделяется исследованию баланса между предсказуемостью совершаемых действий и безопасностью правил, гибкостью понимания естественного языка в больших языковых моделях (LLM), подверженных галлюцинациям и prompt-атакам. Для сравнения реализованы упрощённые прототипы всех трёх архитектур и проведён эксперимент на одном и том же наборе русскоязычных команд с многократными повторениями. Оценивались точность работы, корректность выполнения допустимых команд, обоснованность отказа на недопустимые команды, время отклика и устойчивость к вариативности формулировок, что позволило оценить эффективность каждой архитектуры.

Ключевые слова: *большие языковые модели (LLM), гибридная архитектура, нейро-символьный подход, системы поддержки принятия решений (СППР), IoT-система.*

DOI: 10.21667/1995-4565-2026-95-130-142

Введение

Системы поддержки принятия решений (СППР) применяются для помощи в управлении сложными процессами в условиях неопределённости [1, 2]. Традиционные СППР, работающие на основе строго формализованных правил обработки запросов, обеспечивают предсказуемость результатов [3, 4], но требуют строгого соответствия команд заданным шаблонам. Малейшее отклонение в формулировке приводит к тому, что запрос распознаётся некорректно [1, 5], а поддержание большой базы правил требует значительных вычислительных ресурсов в динамично меняющейся среде [3, 4].

Появление больших языковых моделей (LLM) открыло новые возможности для СППР [2, 6], поскольку LLM способны интерпретировать разнообразные команды на естественном языке без привязки к шаблонам, что повышает гибкость и удобство системы. Например, модель распознает перефразированные или неполные команды, которые не смогла бы понять система, работающая на основе формализованных правил. Однако непосредственное применение LLM для управления устройствами несёт риски надёжности и безопасности, так как генеративная модель может уверенно выдавать некорректные или уязвимые для системы инструкции [6-7]. В частности, в системах управления IoT-устройствами LLM может предложить действие для несуществующего устройства или решения, нарушающие безопасность всей системы [8-9]. Кроме того, отсутствие логического контроля делает систему уязвимой к атакам через вредоносные подсказки (prompt injection), скрывающие в запросе несанкционированные инструкции.

Современные исследования повышают надёжность LLM-агентов за счёт интеграции формальной логики и правил знаний. Логические проверки и ограничения, добавленные к LLM, уменьшают частоту ошибок и нежелательных действий [10, 11]. В гибридных архитектурах языковая модель генерирует возможное решение, а специальный модуль правил проверяет его

корректность перед выполнением [12, 13]. Такой подход блокирует возможные «галлюцинации» модели и обеспечивает предсказуемость поведения [10, 14]. Тем не менее, существующие решения пока не достигают одновременно широкого понимания языка и строгих гарантий безопасности в контексте рассмотрения многоуровневых мультиагентных архитектур.

Цель настоящей **работы** – провести сравнительный анализ различных LLM-ориентированных архитектурных решений и оценить их эффективность при разработке СППР для распределенных систем.

Гипотеза исследования состоит в том, что комбинирование LLM-моделей с механизмами правил позволяет повысить качество поддержки принятия решений за счёт объединения гибкости генеративных моделей и предсказуемости формальных правил.

Анализ существующих подходов

Правило-ориентированные СППР используют формализованную базу знаний, содержащую набор логических правил. Процесс принятия решений строится на механизме логического вывода, сопоставляющего входные данные с этими правилами. Каждая допустимая ситуация или команда должна быть описана заранее, и система строго следует этим предписаниям [3, 4]. Преимущество такого подхода заключается в предсказуемости и объяснимости, поскольку на каждое решение влияет конкретное правило, что упрощает анализ и вызывает доверие оператора.

Такая система также не выполнит запрещённое действие, если команда не предусмотрена или нарушает ограничения. Основной недостаток – низкая гибкость и масштабируемость [4, 11], поскольку даже лишнее слово может сделать понятную команду нераспознаваемой [1, 15, 16]. Кроме того, по мере расширения области применения экспоненциально растёт число правил, требуя постоянного обновления базы знаний экспертом, что затрудняет использование системы в динамичных условиях.

Архитектуры на основе LLM предлагают альтернативный подход к интерпретации команд. Обученная на массиве текстов, LLM понимает разнообразные формулировки запросов и генерирует осмысленный ответ или план действий [6, 7].

Пользователь может взаимодействовать на естественном языке, не соблюдая жёсткий синтаксис команд, а модель выделяет его намерение и возвращает соответствующее действие. В практическом применении LLM-агент формирует ответ на любой однозначно сформулированный запрос [17].

Такой подход делает систему гораздо удобнее для пользователя по сравнению с правило-ориентированной. Однако он сопряжён с существенными рисками, поскольку модель не имеет встроенного механизма проверки и может с одинаковой уверенностью предложить как верное, так и ошибочное решение.

Экспериментально показано, что LLM-агент формирует ответы на запросы, выходящие за пределы фактически доступной функциональности целевой системы, включая сообщения о выполнении несуществующих операций – проявление эффекта галлюцинаций [10, 17]. Подобные искажения особенно критичны при переносе в контур исполнения, где они способны генерировать неправильные для архитектуры действия [8, 9]. Отсутствие прозрачности в работе механизма вывода препятствует проверке корректности решений, ограничивает доверие к системе и усложняет предотвращение возможных сбоев [21, 22].

Гибридные подходы. Для объединения сильных сторон правил и нейросетей предлагается гибридная архитектура, сочетающая LLM с логическим контролем [12, 14]. В гибридной архитектуре LLM используется для семантического анализа входного запроса, тогда как модуль правил осуществляет проверку допустимости и отвечает за выполнение операции.

Сначала команда пользователя на естественном языке обрабатывается LLM для определения намерения, и результат переводится во внутреннюю структурированную форму. Затем эта команда поступает в логический модуль, который содержит явные знания о предметной области, включая существующие устройства, допустимые действия и действующие ограничения.

Модуль правил верифицирует результат работы LLM и обеспечивает выполнение только допустимых операций [12]. Таким образом, система сочетает гибкость языковой модели с надёжностью правил, а нейро-символические системы такого рода демонстрируют уменьшение эффекта «галлюцинаций» и ошибок [10, 14]. Пользователь при этом продолжает общаться на естественном языке, не ограничиваясь заранее заданными шаблонами. Недостатками гибридного подхода являются временные расходы и усложнение системы, но эти издержки оправданы ввиду повышения безопасности и предсказуемости [14].

Предлагаемая в рамках данной работы гибридная архитектура СППР включает два основных компонента – модуль интерпретации на базе LLM и модуль логического вывода (правил). Работа системы осуществляется по следующему алгоритму:

1. **Интерпретация запроса через LLM.** Пользовательская команда на естественном языке поступает в языковую модель, настроенную на контекст IoT-системы. LLM анализирует неструктурированный текст и пытается определить намерение пользователя, формируя внутреннюю структурированную команду. Даже если фраза сформулирована нестандартно, LLM сопоставляет её с известными действиями. Таким образом, LLM выступает в роли «понимающего слушателя», переводя разнообразные выражения пользователя в единый формат для бизнес-логики системы.

2. **Логическая проверка и выполнение.** Сгенерированная LLM команда передаётся в модуль правил, содержащий явные знания о системе. Модуль проверяет существование указанного устройства, разрешённость запрашиваемого действия в текущих условиях и соблюдение ограничений безопасности. При успешном прохождении всех проверок модуль инициирует выполнение команды, где посылает сигнал реальному устройству и формирует пользователю подтверждение. В случае выявления нарушений команда отклоняется. Таким образом, логический модуль служит защитным механизмом для LLM, предотвращая потенциально ошибочные решения модели.

3. **Обратная связь и коррекция.** Архитектуру можно дополнить механизмом обучения на основе обратной связи. Если модуль правил часто отклоняет определённые команды, то это сигнализирует о пробелах в знаниях LLM или новых шаблонах запросов, что определяет необходимость дообучения модели и пополнения базы правил. В текущем прототипе эта функция не реализована, однако в масштабируемых системах обратная связь играет важную роль для улучшения многоуровневой мультиагентной системы.

Благодаря сочетанию этих уровней гибридная система способна интерпретировать широкий спектр пользовательских запросов с помощью LLM и при этом гарантирует отсутствие опасных или некорректных для исполнения действий. В сравнении с правило-ориентированными СППР, гибридная архитектура легко адаптируется к новым формулировкам, а в отличие от чистого LLM-агента включает механизм проверки и предотвращения некорректных операций. Подобный нейро-символьный подход набирает популярность [10, 14], поскольку сочетает статистическое обучение и дедуктивную логику.

В настоящем исследовании прототип гибридной системы реализован в упрощённой форме, где роль LLM выполняет функция на Python, возвращающая заранее заданную структурированную команду по входной фразе, а логический модуль осуществляет проверку команды по списку разрешённых действий. Для полноты сравнения рассмотрены три альтернативные архитектуры.

Правило-ориентированная СППР (вариант А). Классическая система оперирует фиксированным набором разрешённых команд. Пользовательский запрос строго сопоставляется с шаблонами из базы правил. В случае отсутствия точного совпадения система отказывает, а при успешном распознавании команды, соответствующей правилам, она немедленно выполняется. Такой вариант практически исключает ошибки и обеспечивает минимальное время отклика. Удобство использования в такой системе минимальное, поскольку пользователь должен знать поддерживаемые команды и чётко их формулировать. На рисунке 1 изображена диаграмма правило-ориентированной СППР.

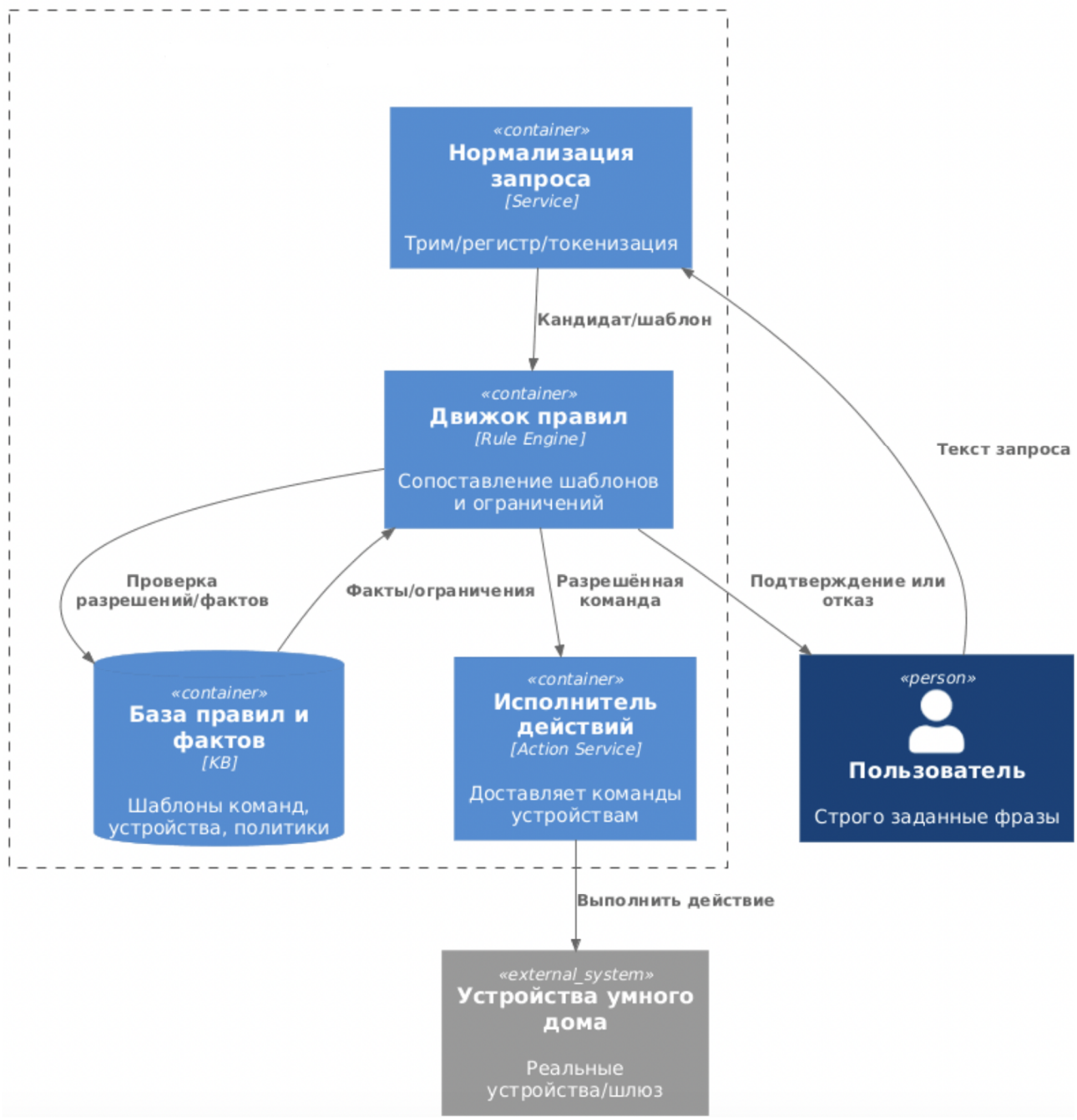

Рисунок 1 – Правило-ориентированная СППР
Figure 1 – Rule-oriented decision support system

LLM-ориентированная СППР (вариант B). Запрос пользователя напрямую подаётся в LLM, которая на основе заложенных знаний генерирует ответ или действие. Предполагается, что модель осведомлена о доступных устройствах в IoT-системе и самостоятельно принимает решение о реакции. Такой агент максимально гибок в понимании языка и попытается интерпретировать даже необычные или некорректно сформулированные команды. Поскольку отдельный слой проверки отсутствует, все сгенерированные моделью действия выполняются напрямую. Это приводит к тому, что LLM иногда генерирует несуществующие или запрещённые действия, будучи «уверенной» в их корректности. На рисунке 2 изображена диаграмма LLM-ориентированной СППР.

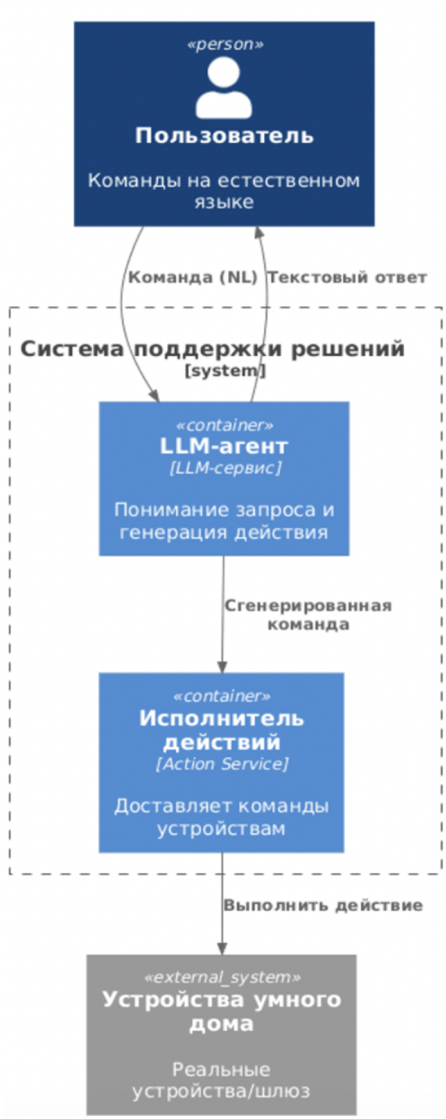

Рисунок 2 – СППР на одной LLM
Figure 2 – DSS for one LLM

Методика исследования

Сценарий и тестовые команды. В качестве предметной области выбрана типовая IoT-система с ограниченным набором устройств. Набор тестовых команд формировался на основе перечня реализованных в системе функций и анализа типовых пользовательских формулировок управления устройствами умного дома. Всего было сформировано 10 русскоязычных команд, включающих 5 допустимых запросов, соответствующих поддерживаемым действиям системы, и 5 недопустимых запросов, выходящих за пределы её функциональности (несуществующие устройства, запрещённые операции или логически невозможные действия). Для оценки устойчивости к языковой вариативности для допустимых команд дополнительно рассматривались перефразированные варианты, содержащие разговорные обороты, избыточные слова и незначительные отклонения формулировки. Такой способ формирования выборки обеспечивает воспроизводимость эксперимента и позволяет объективно оценить способность архитектур корректно выполнять разрешённые действия и обоснованно отклонять недопустимые запросы.

Для оценки эффективности системы использовалась метрика доли корректных исходов (accuracy) [18]. Под корректным исходом понимается либо выполнение допустимой команды, соответствующей функциональности системы, либо обоснованный отказ при недопустимом запросе.

По результатам 900 тестовых запусков, включающих 10 команд, каждая из которых была выполнена 30 раз в рамках трёх архитектур, получены следующие значения точности: для правило-ориентированной архитектуры доля корректных исходов составила 0,80; для LLM-ориентированной – 0,70; для гибридной – 1,00. Снижение точности в варианте А обусловлено отказами при вариативных формулировках допустимых команд, тогда как в варианте В основная доля ошибок связана с выполнением недопустимых действий вследствие эффекта галлюционирования модели. Гибридная архитектура обеспечила корректную обработку всех запросов, включая отклонение недопустимых инструкций.

Для оценки времени отклика измерялось время от подачи команды до получения ответа системы. Помимо среднего времени ответа вычислялся 95-й перцентиль (p95) длительности [19], который показывает, что 95 % запросов обрабатываются не дольше этого времени, а только 5 % самых медленных превышают порог. Эта метрика отражает «хвост» распределения задержек и гарантированный уровень отзывчивости под нагрузкой. В индустрии p95 и p99 зачастую важнее среднего, так как редкие, но сильные задержки заметно влияют на пользовательский опыт [20]. Для каждой архитектуры сравнивались как среднее время, так и p95, что позволило оценить не только типичную скорость, но и стабильность работы, а также учитывать вариативность времени ответа между запросами.

Устойчивость к вариативности формулировок оценивалась на перефразированных вариантах допустимых команд (включая лишние слова, разговорные обороты и незначительные опечатки). В правило-ориентированной архитектуре корректно распознано 12 из 30 вариативных формулировок (40 %), поскольку система требовала строгого совпадения с шаблоном. LLM-ориентированная архитектура продемонстрировала 28 из 30 корректных интерпретаций (93%), однако в 3 случаях зафиксированы ложные срабатывания на недопустимые запросы. Гибридная архитектура обеспечила 29 из 30 корректных интерпретаций (97 %) при полном отсутствии ложных выполнений запрещённых действий, так как логический модуль блокировал некорректные решения модели. Полученные результаты подтверждают, что добавление формального слоя контроля позволяет сохранить высокую языковую гибкость без снижения безопасности исполнения.

Реализация и проведение эксперимента. Для сравнения была разработана программная симуляция всех трёх архитектур. Реализованы три режима работы: А – **правило-ориентированная**, В – **LLM-ориентированная**, С – гибридная в едином окружении. Отличие между режимами заключается исключительно в логике обработки команд.

Вариант А (правило-ориентированная) реализован через прямое сопоставление входной строки с заранее определёнными шаблонами команд. Если после небольшой нормализации текста найдено точное совпадение с шаблоном, то вызывается соответствующая функция устройства. Время выполнения при этом минимально ~0,005 с, так как вычислительная нагрузка мала. Если совпадения нет, то система возвращает отказ.

Вариант В (LLM-ориентированная) смоделирован функцией, которая с условной задержкой ~0,5 с (с колебаниями ±0,1 – 0,2 с) возвращает ответ модели. Заранее было задано, какой ответ LLM должна выдавать на каждую тестовую команду и будет ли он корректным или содержащим «галлюцинацию», чтобы имитировать типичное поведение большой модели. Например, на неподдерживаемый запрос симулятор LLM возвращал успешный ответ о выполнении действия даже в случаях отсутствия подобного устройства.

Вариант С (гибридная) является комбинированным решением, которое вызывает сначала парсинг LLM, что приводит к задержке ~0,5 с для получения структурированной команды, затем проверяет её через модуль правил. Если полученная команда присутствует в списке разрешённых действий, то она выполняется, если нет, то возвращается отказ. Общее время отклика составило ~0,6 – 0,7 с., из которых большая часть обусловлена работой LLM, а проверка занимает лишь несколько миллисекунд. На рисунке 3 изображена диаграмма гибридной СППР.

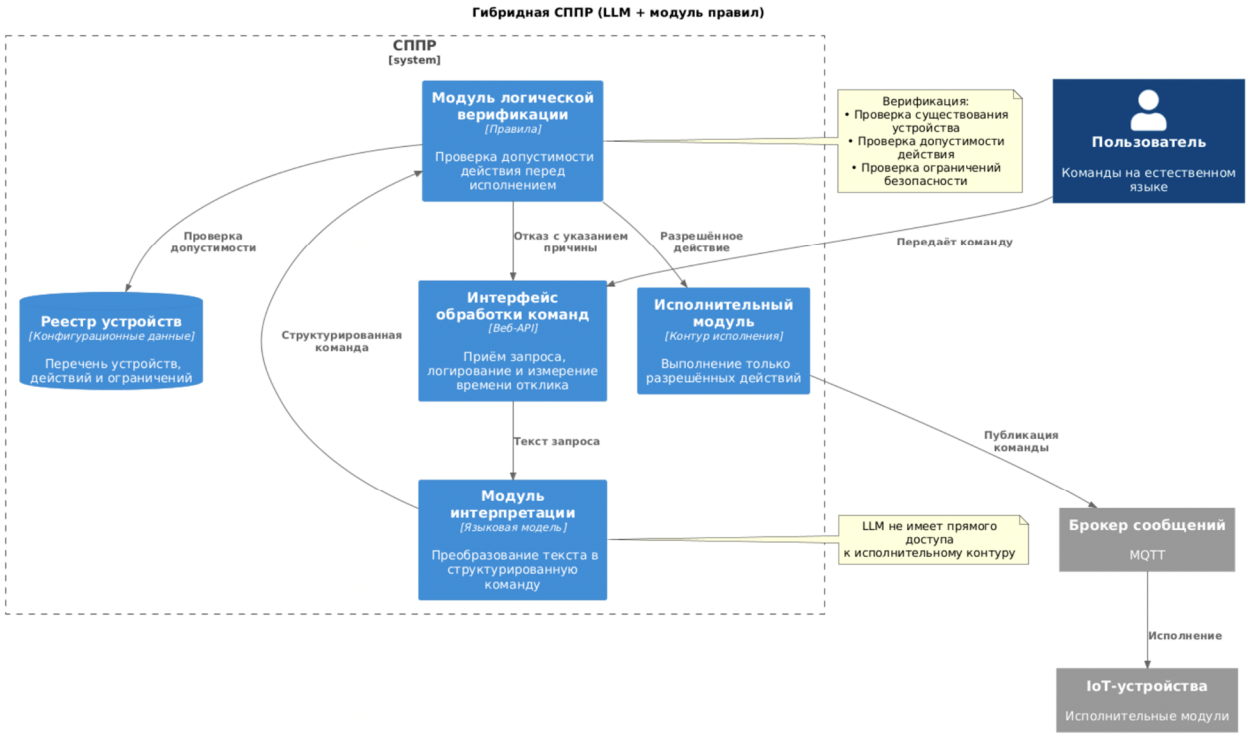

Рисунок 3 – Гибридная СППР (LLM + Правила)
Figure 3 – Hybrid DSS (LLM + Rules)

Кроме того, в ходе проведенных экспериментов оценивались корректность исхода (выполнение допустимой команды либо обоснованный отказ) и время отклика системы.

По результатам измерений для правило-ориентированной архитектуры среднее время отклика составило 0,005 с, p95 – 0,01 с, доля корректных исходов – 0,80. Для LLM-ориентированной архитектуры среднее время составило 0,55 с, p95 – 0,90 с, доля корректных исходов – 0,70. Для гибридной архитектуры среднее время отклика составило 0,70 с, p95 – 1,00 с, при этом доля корректных исходов достигла 1,00.

Многократное повторение эксперимента позволило минимизировать влияние случайных колебаний времени обработки и обеспечить устойчивость полученных результатов.

Результаты и обсуждение

При сравнении оценивалась способность системы корректно выполнять разрешённые команды, отклонять недопустимые и интерпретировать различные формулировки. Полученные результаты подтвердили характерные преимущества и ограничения каждой архитектуры.

Правило-ориентированная система показала наивысшую скорость и надёжность в рамках заложенных сценариев. Среднее время ответа составило около 0,005 с, p95 ~ 0,01 с, что соответствует практически мгновенной реакции. Полученный результат обусловлен низкой вычислительной сложностью алгоритма, сводящегося к прямому сопоставлению входной строки с заранее заданными шаблонами. В ходе эксперимента не было зафиксировано ни одного ложного выполнения действия. Система либо корректно исполняла предусмотренные правилами команды, либо возвращала отказ при отсутствии соответствующего шаблона в базе знаний. Все поддерживаемые системой команды были выполнены корректно, а все недопустимые запросы были отклонены в соответствии с заданной логикой. Остальные случаи были отказами на команды, которые были корректны по смыслу, но сформулированы неточно относительно фраз, хранящихся в базе.

Любое отклонение формулировки приводило к тому, что система не смогла распознать запрос, даже если имелось соответствующее правило. Такие ситуации снижают общую точность, поскольку система не выполняет некоторые разрешённые действия, если они заданы нестандартно. Тем не менее, всех ложных срабатываний удалось избежать, поскольку архи-

тектура на правилах гарантированно не выходит за пределы предусмотренных действий. Для пользователя данный подход определяет достаточно надёжное, объяснимое, но сложное решение, требующее хранения правильных команд.

LLM-ориентированная система продемонстрировала обратный результат. Она корректно интерпретировала почти все пользовательские запросы, независимо от вариаций формулировки. Все осмысленные вводы получили корректный ответ, а гибкость понимания была максимальной. Там, где система на правилах отказала из-за неточного словаря, LLM-агент успешно распознал намерение и выдал правильное действие.

Однако расширение универсальности восприятия было достигнуто ценой снижения точности выполняемых действий. Без контролирующего слоя модель выполнила ряд некорректных команд на запросы, которые не поддерживались. В частности, на недопустимые инструкции LLM сгенерировала ответ на невыполненное действие, то есть «галлюцинировала» выполнение несуществующей функции. В результате эксперимента доля корректных исходов для LLM-ориентированной архитектуры составила 0,70, что ниже показателя правило-ориентированного решения (0,80). Снижение точности обусловлено не ошибками интерпретации намерения пользователя, а выполнением недопустимых действий: в 30 % случаев система сгенерировала и «подтвердила» выполнение операций, отсутствующих в функциональности модели предметной области. Таким образом, основным источником ошибок выступили ложные выполнения, а не отказы при допустимых запросах.

Кроме того, время отклика LLM заметно больше: в среднем ~0,5 – 0,6 с, а p95 доходил до ~0,9 с. Для диалогового режима это приемлемо, но стабильность ниже, чем у правил, иногда задержка достигала почти 1 с. При более сложных запросах или дополнительной сетевой задержке время отклика может увеличиваться, поэтому показатель p95 принципиально важен, поскольку отражает характерные значения задержки в наихудших сценариях, вплоть до почти секундного ожидания. Тем не менее LLM-подход крайне удобен в использовании и потенциально адаптируем под новые задачи без перепрограммирования. Его основные проблемы – непрозрачность принимаемых решений и риск неверных действий модели. В критичных системах полагаться только на LLM опасно без дополнительных мер контроля, тем более что пользователи меньше доверяют такому агенту из-за непредсказуемых ошибок генерации [21, 22].

Гибридная система (LLM + правила) Гибридная архитектура продемонстрировала наилучшие значения по совокупности рассмотренных показателей. Доля корректных исходов составила 1,00, где система выполнила все допустимые команды и отклонила все недопустимые запросы. Ложных выполнений и необоснованных отказов зафиксировано не было. В каждом случае отказа причина была связана исключительно с отсутствием соответствующего действия в функциональности системы либо с установленными ограничениями безопасности.

Этот результат согласуется с литературными данными, где сочетание языковой модели с формальной логикой позволяет блокировать ошибки LLM и гарантировать корректное поведение агента [10, 14]. В тестах настоящей работы логический модуль успешно отфильтровывал некорректные решения модели. Например, на некорректный запрос система сначала идентифицировала намерение, но затем модуль правил обнаружил отсутствие соответствующего устройства и выдал отказ вместо выполнения. Таким образом предотвращалась «галлюцинация» модели, и пользователь получал понятное сообщение о невозможности команды вместо иллюзии выполнения, как это было бы в LLM-ориентированной СППР [10, 17].

По уровню гибкости понимания гибридная архитектура практически не уступала чистой LLM, поскольку начальный этап обработки совпадает, и система распознавала различные формулировки запросов. Время отклика гибридной системы слегка возросло по сравнению с LLM и достигло среднего времени ~0,7 с, при этом p95 ~1,0 с. Добавление проверки правил увеличило задержку всего на 0,1 – 0,2 с, что незаметно для пользователя. Основную часть времени по-прежнему занимала работа LLM, а логический вывод составлял доли секунды.

Даже если правила усложнить, их влияние на общую задержку не приведет к значительным изменениям по скорости ответа. Согласно другим исследованиям, полный нейро-символьный агент работал лишь на несколько процентов медленнее чистого LLM [14]. В наших измерениях разница p95 составила десятую долю секунды 1,0 с против 0,9 с, то есть издержки в виде безопасности небольшой задержки минимальны. В таблице 1 описаны обобщены ключевые различия архитектур. Стенд был развёрнут на вычислительном узле архитектуры x86-64, оснащённом процессором Intel Core i7-13700K (16 ядер), графическим ускорителем NVIDIA RTX 4060, 32 ГБ оперативной памяти DDR4 и SSD-накопителем объёмом 1 ТБ.

Таблица 1 – Ключевые различия архитектур
Table 1 – Key differences between architectures

	Варианты исполнения архитектуры СППР		
Критерий	**A**	**B**	**C**
Среднее время отклика, с	0,005	0,55	0,70
p95, с	0,01	0,90	1,00
Доля корректных исходов (accuracy)	0,80	0,70	1,00
Доля ложных выполнений	0	0,30	0
Объяснимость решений	Высокая (каждый шаг предопределён правилом)	Низкая («чёрный ящик»)	Средняя (логический модуль сообщает причину отказа)
Сложность реализации	Низкая/Средняя (ручная разработка базы знаний)	Средняя (создание и настройка модели)	Высокая (нужны и модель, и правила, и их интеграция)

По результатам приведенного анализа в таблице 1 видно, что лишь архитектура C одновременно обеспечивает высокую гибкость и точность работы, сводя компромиссы к минимуму. Практически это означает, что такая система подходит для промышленных приложений, где важны удобство для пользователя и гарантии безопасности. Небольшое увеличение задержки ответа до ~1 с вполне допустимо в диалоговом взаимодействии, учитывая выигрыш в надёжности.

В дальнейшем стоит оптимизировать компонент LLM для снижения времени ответа без потери качества понимания, чтобы приблизить задержки к реальному времени. Уже сейчас добавление логического контроля заметно повышает доверие к системе, где пользователь может быть уверен, что агент не совершит некорректных действий, а отказ будет обоснованным, а не вызванным непониманием системы. Ожидается, что доверие пользователей к гибридным решениям выше, так как они более прозрачны и предсказуемы, чем полностью нейросетевые [21, 22].

Заключение

В работе представлен прототип интеллектуальной СППР для управления устройствами IoT-систем, объединяющей большую языковую модель и модуль логических правил. Реализованы три варианта архитектуры – правило-ориентированной, LLM-ориентированный и гибридной, а также проведено их сравнение по метрикам точности выполнения команд, устойчивости к вариациям ввода, скорости ответа, включая p95 и другие показатели.

Результаты показывают, что классическая система на правилах обеспечивает минимальные задержки и полностью предсказуемое поведение, но требует строго заданных формулировок команд, что ограничивает её применение.

Подход с LLM-ориентированной архитектурой значительно повышает гибкость и естественность взаимодействия, однако без контроля качество исполнения снижается ввиду возможных «галлюцинаций» и ошибок.

Гибридная архитектура LLM + правила продемонстрировала наилучшее суммарное качество, поскольку она надёжно интерпретирует различные запросы и выполняет только разрешённые действия. Логический модуль устраняет ошибки модели, почти не влияя на быстродействие системы. Таким образом, гибридный подход обеспечивает одновременно гибкость и безопасность, повышая надёжность СППР на базе LLM при минимальном влиянии на время отклика.

Комбинация большой языковой модели с формальными методами представляется перспективным направлением для создания надёжных интеллектуальных агентов. Такая архитектура сохраняет преимущества нейросетевого искусственного интеллекта и дополняет их механизмами логического контроля, придавая системе необходимую ответственность и объяснимость решений. Полученные результаты согласуются с представлениями о модульной организации СППР на основе продукционных правил, где подчёркивается важность явной структуры базы знаний и прозрачного механизма вывода [25]. Это особенно важно в сферах, где цена ошибки высока, пользователям и системам нужна уверенность, что ИИ не выйдет за рамки допустимого поведения [23, 24].

Настоящее исследование носит экспериментальный характер и ограничено упрощённым сценарием, но подтверждает эффективность гибридного подхода и даёт ориентиры для дальнейшей работы.

В последующем исследовании планируется расширить архитектуру на более сложные многоагентные сценарии, реализовать обучение с обратной связью, чтобы LLM улучшала понимание на основе отклонённых команд, а также изучить автоматическое извлечение правил из данных. Кроме того, актуальна задача оптимизации LLM-модуля, например, использование облегчённых локальных моделей или повышение эффективности инференса, чтобы сократить время отклика без потери качества.

Отдельно предполагается развивать средства объяснения решений оператору, предоставлять понятную обратную связь о причине отказа или принятого действия для поддержания доверия пользователя.

Таким образом, показано, что нейро-символьная гибридная архитектура может лечь в основу нового поколения СППР и умных помощников, совмещающих адаптивность и безопасность. Объединение статистических методов ИИ с формальными гарантиями позволяет создать системы, способные учиться и рассуждать, но при этом остающиеся под контролем разработчика в ключевых точках принятия решений. Такие агенты потенциально найдут более широкое применение в промышленности и повседневной жизни, сочетая лучшие черты «классического» ИИ и современных моделей машинного обучения.

Библиографический список

1. **Блюмин С.Л., Шуйкова И.А.** Модели и методы принятия решений в условиях неопределённости. Липецк: ЛЭГИ, 2001. 138 с.

2. **Рассел С., Норвиг П.** Искусственный интеллект: современный подход. 3-е изд. М.: Вильямс, 2016. 1408 с.

3. **Vassilakopoulos M.** Strengths and Weaknesses of LLM-Based and Rule-Based NLP… Electronics, 2025, 14 (15):3064.

4. **Michel-Delétie C., et al.** Neuro-Symbolic Methods for Trustworthy AI: A Systematic Review (2021–2022). Neurosymbolic Artificial Intelligence, 2023.

5. **Чернышев И.В.** Применение локальных языковых моделей в задачах интеллектуальной автоматизации // Искусственный интеллект и принятие решений. 2024. № 2. С. 55-63.

6. **Brown T.B., Mann B., Ryder N., et al.** Language Models are Few-Shot Learners. NeurIPS. 2020. 33:1877-1901.

7. **Bender E.M., Gebru T., McMillan-Major A., Shmitchell S.** On the Dangers of Stochastic Parrots. FAccT ’21, 2021. С. 610-623.

8. **Dang A.-H., Tran V., Nguyen L.-M.** Survey and Analysis of Hallucinations in LLMs. Frontiers in AI, 2025, 8:1622292.

9. **Liu Y., Deng G., Li Y., et al.** Prompt Injection Attack Against LLM-Integrated Applications. arXiv:2306.05499, 2023.

10. **Nawaz U., et al.** A Review of Neuro-Symbolic AI Integrating Reasoning and Learning. AI Open, 2025 (in press).

11. **Rezunik L., Prozorskiy M.A., Alexandrov D.V.** Combining Logical Reasoning and LLMs Toward Creating Multi-Agent Smart Home Systems. Proc. ISP RAS, 2025, 37(4-2):219-234.

12. **Bollikonda M.** Hybrid AI Reasoning: Integrating Rule-Based Logic with Transformer Inference. Preprints.org, 2025, препринт № 202504.1453 (v. 1).

13. **Li S., Guo Y., Yao J., Liu Z., Wang H.** HomeBench: Evaluating LLMs in Smart Homes… ACL 2025 (Long Papers), 2025 (in press); препринт: ACL Anthology 2025.acl-long.597.

14. **Конев А.А., Паюсова Т.И.** Bol'shie yazykovye modeli v IB i penteste: sistematicheskij ob-zor. NTV ITMO, 2025, 25(1):42-52.

15. **Zheng Y., et al.** A Review on Edge Large Language Models. ACM Computing Surveys, 2025 (online first).

16. **Luger E., Sellen A.** Like having a really bad PA: The gulf between user expectation and experience of conversational agents. B: CHI 2016. ACM. 2016, pp. 5286-5297.

17. **Brown T.B., Mann B., Ryder N., et al.** Language Models are Few-Shot Learners. NeurIPS, 2020. 33:1877-1901.

18. **Li S., Guo Y., Yao J., Liu Z., Wang H.** HomeBench: Evaluating LLMs in Smart Homes… ACL 2025. 2025 (in press).

19. **Dean J., Barroso L.A.** The Tail at Scale. Communications of the ACM, 2013, 56(2):74-80.

20. **Misra P.A., Borge M.F., Goiri I., et al.** Managing Tail Latency in Datacenter-Scale File Systems… EuroSys 2019. ACM. 2019.

21. **Afroogh S., et al.** Trust in AI: Progress, Challenges, and Future Directions. Humanities and Social Sciences Communications. 2024, 11:1568.

22. **Liu X., et al.** Automatic and Universal Prompt Injection Attacks Against LLMs. arXiv: 2403.04957. 2024.

23. **Michel-Delétie C., et al.** Neuro-Symbolic Methods for Trustworthy AI Neurosymbolic Artificial Intelligence. 2023.

24. **Nawaz U., et al.** A Review of Neuro-Symbolic AI Integrating Reasoning and Learning. AI Open, 2025 (in press).

25. **Сорокин А.Б., Железняк Л.М., Супруненко Д.В., Холмогоров В.В.** Проектирование модулей системной динамики в системах поддержки принятия решений // Russian Technological Journal. 2022. Т. 10. № 4. С. 18-26. DOI: 10.32362/2500-316X-2022-10-4-18-26.

UDC 004.89:005.53

COMPARATIVE ANALYSIS OF ARCHITECTURES OF DECISION SUPPORT SYSTEMS BASED ON RULES, LARGE LANGUAGE MODELS, AND THEIR HYBRID COMBINATIONS

A. O. Kostyrenkov, PhD Student, Assistant Professor, Department of Instrumentation and Applied Software, IIT RTU MIREA, Moscow, Russia;
orcid.org/0009-0007-0294-694X, e-mail: kostyrenkov@mirea.ru
E. N. Miftakhov, Doctor in Physics and Mathematics, professor, Department of Instrumentation and Applied Software, IIT RTU MIREA, Moscow, Russia;
orcid.org/0000-0002-0471-5949, e-mail: promif@mail.ru

This paper presents a comparative analysis of three decision support system (DSS) architectures for device control: rule-based, LLM-based, and hybrid (LLM + rules). Particular attention is paid to exploring the balance between the predictability of actions and the security of rules, as well as the flexibility of natural

language understanding in large language models (LLM) susceptible to hallucinations and prompt attacks. For comparison, simplified prototypes of all three architectures were implemented, and an experiment was conducted on the same set of Russian-language commands with multiple repetitions. The accuracy of operation, correct execution of valid commands, the justification for rejecting invalid commands, response time, and resilience to wording variability were assessed, allowing us to evaluate the effectiveness of each architecture.

Keywords: *Large language models (LLM), hybrid architecture, neuro-symbolic approach, decision support systems (DSS), IoT system.*

DOI: 10.21667/1995-4565-2026-95-130-142

References

1. **Blumin S.L., Shuykova I.A.** *Models and Methods of Decision Making under Uncertainty*. Lipetsk: LEGI, 2001. 138 p.
2. **Russell S., Norvig P.** *Artificial Intelligence: A Modern Approach*. 3rd ed. Moscow: Williams, 2016. 1408 p.
3. **Vassilakopoulos M.** *Strengths and Weaknesses of LLM-Based and Rule-Based NLP…* Electronics, 2025, 14 (15):3064.
4. **Michel-Delétie C., et al.** Neuro-Symbolic Methods for Trustworthy AI: A Systematic Review (2021-2022). *Neurosymbolic Artificial Intelligence*. 2023.
5. **Chernyshev I.V.** Application of Local Language Models in Intelligent Automation Problems. *Artificial Intelligence and Decision Making*. 2024, no. 2, pp. 55-63.
6. **Brown T.B., Mann B., Ryder N., et al.** Language Models are Few-Shot Learners. *NeurIPS*, 2020. 33:1877-1901.
7. **Bender E.M., Gebru T., McMillan-Major A., Shmitchell S.** On the Dangers of Stochastic Parrots. *FAccT '21*. 2021, pp. 610–623.
8. **Dang A.-H., Tran V., Nguyen L.-M.** Survey and Analysis of Hallucinations in LLMs. *Frontiers in AI*. 2025, 8:1622292.
9. **Liu Y., Deng G., Li Y., et al.** Prompt Injection Attack Against LLM-Integrated Applications. *arXiv*: 2306.05499, 2023.
10. **Nawaz U., et al.** A Review of Neuro-Symbolic AI Integrating Reasoning and Learning. *AI Open*. 2025 (in press).
11. **Rezunik L., Prozorskiy M.A., Alexandrov D.V.** Combining Logical Reasoning and LLMs Toward Creating Multi-Agent Smart Home Systems. *Proc. ISP RAS*. 2025, 37(4-2):219-234.
12. **Bollikonda M.** Hybrid AI Reasoning: Integrating Rule-Based Logic with Transformer Inference. Preprints.org, 2025, preprint no. 202504.1453 (v. 1).
13. **Li S., Guo Y., Yao J., Liu Z., Wang H.** HomeBench: Evaluating LLMs in Smart Homes… *ACL 2025* (Long Papers). 2025 (in press); preprint: ACL Anthology 2025.acl-long.597.
14. **Konev A.A., Payusova T.I.** Bol'shie yazykovye modeli v IB i penteste: sistematicheskij obzor. NTV *ITMO*. 2025, 25(1):42-52.
15. **Zheng Y., et al.** A Review on Edge Large Language Models. *ACM Computing Surveys*. 2025 (online first).
16. **Luger E., Sellen A.** Like having a really bad PA: The gulf between user expectation and experience of conversational agents. B: CHI 2016. *ACM*. 2016, pp. 5286-5297.
17. **Brown T.B., Mann B., Ryder N., et al.** Language Models are Few-Shot Learners. *NeurIPS*. 2020, 33:1877-1901.
18. **Li S., Guo Y., Yao J., Liu Z., Wang H.** HomeBench: Evaluating LLMs in Smart Homes… *ACL 2025*. 2025 (in press).
19. **Dean J., Barroso L.A.** The Tail at Scale. *Communications of the ACM*. 2013, 56(2):74-80.
20. **Misra P.A., Borge M.F., Goiri I., et al.** Managing Tail Latency in Datacenter-Scale File Systems… EuroSys 2019. *ACM*, 2019.
21. **Afroogh S., et al.** Trust in AI: Progress, Challenges, and Future Directions. *Humanities and Social Sciences Communications*. 2024, 11:1568.
22. **Liu X., et al.** Automatic and Universal Prompt Injection Attacks Against LLMs. *arXiv*: 2403.04957, 2024.

23. **Michel-Delétie C., et al.** *Neuro-Symbolic Methods for Trustworthy AI Neurosymbolic Artificial Intelligence*, 2023.

24. **Nawaz U., et al.** A Review of Neuro-Symbolic AI Integrating Reasoning and Learning. *AI Open*. 2025 (in press).

25. **Sorokin A.B., Zheleznyak L.M., Suprunenko D.V., Kholmogorov V.V.** Design of system dynamics modules in decision support systems. *Russian Technological Journal*. 2022, 10 (4), pp. 18-26. DOI: 10.32362/2500-316X-2022-10-4-18-26.