Введение

В последние годы наблюдается стремительный рост интереса к технологиям искусственного интеллекта и их применению для решения прикладных задач. Одним из наиболее динамично развивающихся направлений является создание интеллектуальных диалоговых систем, или чат-ботов, способных взаимодействовать с пользователем на естественном языке. Современные чат-боты эволюционировали от простых систем, работающих по жестко заданным сценариям, до сложных ассистентов, использующих модели обработки естественного языка (NLP) для понимания намерений пользователя и предоставления релевантной информации.

Особый интерес представляет применение таких систем в узкоспециализированных областях, таких как туризм, экология и образование. Предоставление точной, верифицированной и, что немаловажно, наглядной информации является ключевой задачей для повышения осведомленности и интереса к уникальным природным объектам. Байкальский регион, с его эндемичной флорой и фауной, представляет собой идеальную предметную область для применения подобных технологий. Традиционные источники информации, такие как справочники или веб-сайты, часто не обладают необходимой интерактивностью и не способны отвечать на комплексные, контекстуально-зависимые вопросы пользователей.

Разработка специализированного чат-бота, ориентированного на экосистему Байкала, позволяет решить эту проблему, предоставляя пользователю мощный инструмент для исследования региона. Однако создание такого ассистента сопряжено с рядом технических вызовов. Необходимо не только обеспечить точное распознавание запросов, но и реализовать эффективное взаимодействие с базами данных, содержащими разнородную информацию: текстовые описания, изображения и геопространственные данные. Современный подход к решению таких задач заключается в использовании гибридных архитектур, сочетающих сильные стороны классических NLU-фреймворков и больших языковых моделей (LLM).

Цель проекта:

Разработать Telegram-бота "Эко-ассистент Байкала", способного предоставлять пользователям мультимедийную информацию (текст,

изображения, интерактивные карты) о флоре и фауне Байкальского региона, основываясь на анализе запросов на естественном языке.

Задачи для достижения цели:

1. **Создание структуры базы данных и хранилища файлов** для мультимедийной информации, включая её метаописания.
2. **Формирование множества признаков,** которые будут описывать мультимедийную информацию.
3. **Создание сценариев диалогов,** которые являются комбинацией формата информации, набора признаков и шаблонов многоактовых диалогов.
4. **Разработка требований к исходным данным,** результатом которой станут требования к формату данных и их описанию с учётом множества признаков.
5. **Разработка программных средств для извлечения данных** (создание и тестирование парсеров для источников, таких как iNaturalist и Байкальский музей, получение геокоординат из справочников).
6. **Разработка и программная реализация сценариев классификации** мультимедийных данных согласно множеству признаков на основе расширяемого набора ИИ-моделей.
7. **Разработка программных средств генерации мультимедийных данных** с применением больших языковых моделей (на примере текстовых и геоданных), в том числе с использованием архитектуры RAG (Retrieval-Augmented Generation).
8. **Заполнение базы данных и хранилища файлов.**
9. **Реализация сценариев диалогов** для разных целевых платформ (Telegram).
10. **Протестировать и оценить работоспособность и производительность** реализованной системы.

1. Анализ

1.1 Диалоговые системы и их эволюция

Диалоговые системы, или чат-боты, представляют собой программные комплексы, предназначенные для имитации осмысленного разговора с пользователем. Исторически первые чат-боты, такие как ELIZA (1966), работали на основе простого сопоставления с шаблонами и не обладали реальным "пониманием" языка. С развитием технологий обработки естественного языка (Natural Language Processing, NLP) чат-боты стали значительно сложнее.

Современные системы можно условно разделить на две большие категории:

- **Декларативные (Rule-Based):** Работают по заранее определенным правилам и сценариям. Они предсказуемы, надежны и быстры, но их возможности строго ограничены заложенными в них скриптами. Такие

боты часто используются в службах поддержки для ответов на типовые вопросы.

- **Интеллектуальные (AI-Based):** Используют машинное обучение и NLP для анализа и понимания запросов пользователя. Они способны распознавать намерения (intents), извлекать сущности (entities) и поддерживать более гибкий диалог.

1.2 Архитектура современных NLU-ассистентов

Ключевым компонентом интеллектуального чат-бота является модуль NLU (Natural Language Understanding). Его основная задача — преобразовать неструктурированный текст пользователя в структурированный формат, понятный машине. Стандартный NLU-пайплайн включает в себя несколько этапов:

1. **Токенизация:** Разбиение текста на отдельные слова или символы (токены).
2. **Извлечение сущностей (Entity Extraction):** Распознавание в тексте важных фрагментов, таких как имена, даты, географические названия (например, "кедр", "зимой", "Листвянка").
3. **Определение намерения (Intent Classification):** Классификация всего запроса пользователя для определения его основной цели (например, получить картинку, получить информацию).

Популярным фреймворком для построения таких систем является **Rasa Open Source**. Он предоставляет инструменты для создания NLU-моделей и управления диалогом (Dialogue Management), позволяя разработчику полностью контролировать логику бота.

1.3 Большие языковые модели (LLM) в диалоговых системах

С появлением больших языковых моделей, таких как GPT-4 или GigaChat, произошел качественный скачок в возможностях диалоговых систем. В отличие от классических NLU-моделей, которые обучаются на конкретных примерах для распознавания ограниченного набора интентов и сущностей, LLM обладают "общим" пониманием языка.

В контексте чат-ботов LLM могут использоваться для решения двух основных задач:

1. **Анализ и структурирование запроса:** LLM может выступить в роли "универсального NLU-модуля". Ему можно дать на вход сырой текст пользователя и попросить вернуть структурированный JSON-объект с намерением и извлеченными параметрами. Этот подход обеспечивает невероятную гибкость, так как система способна понимать запросы, которые не были явно предусмотрены в обучающих данных.
2. **Генерация ответа:** LLM может генерировать человекоподобные текстовые ответы, что особенно полезно для фоллбэк-сценариев, когда в основной базе знаний нет готового ответа.

1.4 Архитектура с переключаемыми пайплайнами обработки

Для решения поставленных задач в рамках проекта "Эко-ассистент Байкала" была спроектирована и реализована архитектура, основанная на двух независимых, переключаемых **конвейерах (режимах)** обработки запросов. Такой подход позволяет пользователю самому выбирать способ взаимодействия с системой в зависимости от его потребностей, сочетая сильные стороны различных технологий.

Обоснование наличия двух режимов заключается в следующем: режим на основе LLM (GigaChat) является основным и наиболее гибким, тогда как режим Rasa выступает в качестве **резервного**. Он может быть задействован в случае недоступности LLM по различным причинам, например, при отсутствии финансирования для использования платных API, что обеспечивает непрерывность работы системы.

Система включает в себя следующие режимы:

1. **Режим Rasa:** В данном режиме вся обработка запроса делегируется классическому NLU-фреймворку Rasa Open Source. Этот процесс оптимизирован для распознавания заранее определенных, структурированных команд. Пользовательский ввод проходит через NLU-модель, которая извлекает намерение (intent) и сущности (entities), после чего система выполняет соответствующее действие (action). Этот режим обеспечивает высокую скорость и предсказуемость для стандартных и частотных запросов.
2. **Режим GigaChat:** В этом режиме система использует большую языковую модель (LLM) GigaChat для семантического анализа запроса. Вместо того чтобы сопоставлять запрос с заранее определенными шаблонами, основной бот отправляет сырой текст пользователя в LLM со специальным системным промптом. Задача LLM — не сгенерировать ответ, а **преобразовать неструктурированный запрос в структурированную JSON-команду**, которую затем исполняет бот. LLM определяет намерения пользователя согласно заранее заданным классам (шаблонам) и извлекает параметры для решения задачи, то есть **осуществляет заполнение слотов** в рамках заданных фреймов. Этот подход обеспечивает высокую гибкость и позволяет понимать сложные, вариативные формулировки на естественном языке.

Пользователь может в любой момент переключиться между этими двумя режимами через меню настроек бота. Несмотря на различие в логике обработки, оба пайплайна обращаются к единому бэкенд-сервису для получения фактических данных (текстов, изображений, карт) и могут использовать общий микросервис для фоллбэк-сценариев. Такая архитектура позволяет экспериментально сравнивать эффективность двух различных подходов к созданию диалоговых систем в рамках одного приложения.

1.5 Проблема и актуальность разработки

Проблема: Существующие источники информации о флоре и фауне Байкальского региона (веб-сайты, справочники, общие ассистенты) обладают рядом фундаментальных недостатков. Они либо неинтерактивны, либо не обладают специализированной и верифицированной базой данных, предоставляя общую или неточную информацию (как общие ассистенты, склонные к "галлюцинациям"). Отсутствует единый инструмент, который бы сочетал глубину специализированных знаний с гибкостью современного диалогового интерфейса, ориентированного на **мультимедийный контент,** такой как генерация интерактивных карт ареалов видов, предоставление фотогалерей по сложным запросам или отображение таксономических связей в графическом виде. Ключевая проблема заключается в **обработке научно-обоснованных мультимедийных данных с учетом контекста** диалога.

Актуальность: Разработка "Эко-ассистента Байкала" является актуальной, поскольку напрямую решает обозначенную проблему. Актуальность проекта заключается в необходимости **внедрения современных технологий взаимодействия**, таких как цифровые помощники, для потенциальных посетителей Байкальского музея и всех интересующихся регионом. Проект предлагает не просто чат-бота, а полноценный инструмент для исследования экосистемы Байкала, который позволяет не только получать текстовую информацию, но и взаимодействовать с мультимедийными данными в интерактивном режиме. Ценность проекта подкрепляется его высоким потенциалом в сферах туризма, образования и эко-просвещения.

Обзор существующих программных средств

Для определения уникальности и актуальности разрабатываемого "Эко-ассистента Байкала" необходимо провести анализ существующих на рынке программных решений, которые частично или полностью пересекаются с его функциональностью. Анализ будет проведен по трем ключевым категориям: общие интеллектуальные ассистенты, специализированные приложения-определители и традиционные информационные чат-боты. Оценка будет проводиться по адаптированным критериям, релевантным для диалоговых систем.

2.1 Общие интеллектуальные ассистенты (Яндекс Алиса, Google Assistant)

Данная категория представляет собой наиболее технологически продвинутые диалоговые системы, способные поддерживать разговор на широкий круг тем.

- **2.1.1 Общая оценка интерфейса.** Интерфейс является преимущественно голосовым, но также поддерживает текстовый ввод. Взаимодействие максимально приближено к естественному человеческому диалогу. Системы отлично справляются с поддержанием контекста и обработкой сложных, многосоставных предложений.

- **2.1.2 Объем и структура представленной информации.** Объем информации практически неограничен, так как ассистенты используют для ответов всю проиндексированную сеть Интернет. Однако это является и их недостатком: информация не является специализированной и часто представляет собой краткую выдержку из первого найденного источника (например, Википедии). Структура ответа, как правило, — это короткий текстовый блок, иногда сопровождаемый ссылкой или изображением из поиска. Мультимедийные возможности ограничены показом статичных картинок и не включают генерацию интерактивных карт по запросу.
- **2.1.3 Наличие и структура меню.** Меню как таковое в классическом понимании отсутствует. Навигация осуществляется полностью через языковые команды. Существуют стандартные команды для вызова справки, но нет структурированных меню для навигации по узкоспециализированной предметной области.
- **2.1.4 Удобство форм для ввода информации.** Основной формой ввода является текстовая строка или голосовой запрос. Благодаря использованию мощных LLM, эти системы обладают высочайшей гибкостью в понимании естественного языка, синонимов и различных формулировок одного и того же вопроса.
- **2.1.5 Возможность поиска информации.** Поиск является основной функцией, но он носит общий характер. Ассистенты не подключены к специализированным, верифицированным базам данных, что в контексте научной информации о флоре и фауне может приводить к предоставлению неточной, устаревшей или откровенно ложной информации ("галлюцинациям").
- **2.1.6 Общий вывод по категории.** Общие ассистенты предоставляют лучший на рынке опыт естественного диалога, но не могут служить надежным источником для получения специализированных знаний. Их функциональность не приспособлена для решения узких задач, таких как построение карт ареалов или поиск изображений по сложным признакам.

2.2 Специализированные приложения-определители (Picture This, iNaturalist)

Это мобильные приложения, основная задача которых — идентификация видов растений и животных по фотографии пользователя.

- **2.2.1 Общая оценка интерфейса.** Интерфейс является графическим (GUI), а не диалоговым. Взаимодействие с пользователем происходит через элементы управления: кнопки, вкладки, экраны. Интерфейс интуитивно понятен для своей основной задачи — загрузки и анализа фотографий.
- **2.2.2 Объем и структура представленной информации.** Эти приложения обладают огромными, хорошо структурированными и

верифицированными базами данных. Информация по каждому виду включает таксономию, качественные фотографии, подробные описания и, в случае iNaturalist, карту с точками наблюдений от других пользователей. Данные представлены в формате карточек, что очень удобно для восприятия.

- **2.2.3 Наличие и структура меню.** Навигация осуществляется через стандартные для мобильных приложений меню (например, нижняя панель вкладок), которые позволяют переключаться между функциями: идентификация, личная коллекция, карта и т.д.
- **2.2.4 Удобство форм для ввода информации.** Основной "формой ввода" является камера или галерея устройства. Текстовый ввод используется только для поиска по названию вида и не предполагает обработки запросов на естественном языке. Задать вопрос вроде "какие хвойные деревья растут на берегу Байкала?" невозможно.
- **2.2.5 Возможность поиска информации.** Поиск ограничен либо названием вида, либо идентификацией по фото. Сложные, многокритериальные запросы не поддерживаются.
- **2.2.6 Общий вывод по категории.** Приложения-определители являются превосходным источником достоверной и хорошо структурированной информации. Однако они не являются диалоговыми системами и не способны обеспечить интерактивное исследование предметной области через вопросы на естественном языке.

2.3 Традиционные информационные чат-боты (банковские, справочные)

К этой категории относятся чат-боты, работающие по заранее заданным сценариям и правилам, часто встречающиеся на сайтах компаний для поддержки клиентов.

- **2.3.1 Общая оценка интерфейса.** Интерфейс строго функционален, часто основан на кнопочных меню. Диалог ощущается как "механический" и негибкий.
- **2.3.2 Объем и структура представленной информации.** Объем информации строго ограничен базой знаний, заложенной разработчиками. Ответы представляют собой заранее написанные текстовые блоки.
- **2.3.3 Наличие и структура меню.** Меню является основным элементом навигации. Оно имеет древовидную структуру и позволяет пользователю предсказуемо перемещаться по доступным опциям.
- **2.3.4 Удобство форм для ввода информации.** Понимание естественного языка у таких ботов крайне ограничено или отсутствует. Они хорошо распознают ключевые слова ("доставка", "цена"), но любой отход от ожидаемого формата запроса приводит к ошибке "я вас не понял".

- **2.3.5 Возможность поиска информации.** Поиск основан на простом сопоставлении с ключевыми словами или навигации по базе знаний через меню.
- **2.3.6 Общий вывод по категории.** Традиционные чат-боты надежны и быстры в рамках своих узких, заранее определенных задач. Однако они абсолютно не подходят для роли исследовательского инструмента, так как не обладают гибкостью для понимания разнообразных вопросов пользователя.

Общий вывод

Проведенный анализ существующих программных средств показывает, что на рынке отсутствует решение, которое бы объединяло сильные стороны всех рассмотренных категорий. Общие ассистенты предлагают гибкий диалог, но не имеют доступа к специализированной и достоверной базе данных. Приложения-определители обладают такой базой, но лишены диалогового интерфейса. Традиционные чат-боты слишком ограничены в своих возможностях. Ценность и новизна "Эко-ассистента Байкала" заключается в том, что он спроектирован для заполнения именно этой ниши. Он сочетает гибкость понимания естественного языка, характерную для современных ассистентов, со специализированной, верифицированной базой знаний, дополняя это уникальными мультимедийными возможностями, такими как генерация интерактивных карт по запросу. Это делает его не просто очередным чат-ботом, а полноценным интерактивным инструментом для исследования.

Процесс AS IS vs TO BE

Для детального обоснования актуальности и ценности разрабатываемого "Эко-ассистента Байкала" был проведен анализ бизнес-процессов получения специализированной информации пользователем. С помощью нотации BPMN 2.0 (Business Process Model and Notation) были смоделированы два состояния процесса: текущее, до внедрения системы (AS IS), и целевое, после ее внедрения (TO BE). Сравнение этих моделей наглядно демонстрирует решаемые проблемы и эффективность предлагаемого архитектурного решения.

1 Модель процесса AS IS: Существующий порядок получения информации

Текущий процесс получения пользователем мультимедийной информации об экосистеме Байкала является фрагментированным, неструктурированным и возлагает основную когнитивную нагрузку на самого пользователя. Диаграмма процесса AS IS (Рисунок 1 – BPMN диаграмма процесса AS IS) иллюстрирует данный подход.

2 Описание диаграммы

Процесс, представленный на диаграмме, можно описать следующими этапами:

1. Инициация и выбор инструмента: процесс начинается с возникновения у пользователя информационной потребности. Ключевой проблемой на данном этапе является задача выбора релевантного инструмента. Пользователь вынужден самостоятельно принимать решение об использовании поисковых систем (Google, Yandex), общих интеллектуальных ассистентов (Яндекс Алиса), специализированных мобильных приложений-определителей (iNaturalist) или научных веб-сайтов. Выбор зависит от предполагаемого типа информации (текст, изображение, идентификация вида), что требует от пользователя предварительных знаний о возможностях и ограничениях каждого источника.
2. Поиск и фильтрация: на следующем этапе пользователь выполняет поиск и осуществляет ручную фильтрацию полученных результатов. Этот шаг сопряжен с анализом большого объема неструктурированных данных, включая общие статьи, блоги и форумы, которые часто не обладают необходимой верификацией.
3. Синтез данных: наиболее трудоемким и неэффективным этапом является задача ручного сопоставления и синтеза информации из различных, не связанных между собой источников. Для получения ответа на комплексный, многокритериальный запрос (например, "где рядом с мысом Хобой можно встретить цветущий эдельвейс?") пользователь должен самостоятельно объединить текстовые описания ареалов, данные с картографических сервисов и релевантные изображения.
4. Результат: исход процесса непредсказуем. Как показывает шлюз на диаграмме, поиск часто либо прекращается неудачей, либо приводит к получению неполного или неточного ответа, лишенного необходимой мультимедийной наглядности (например, интерактивной карты).

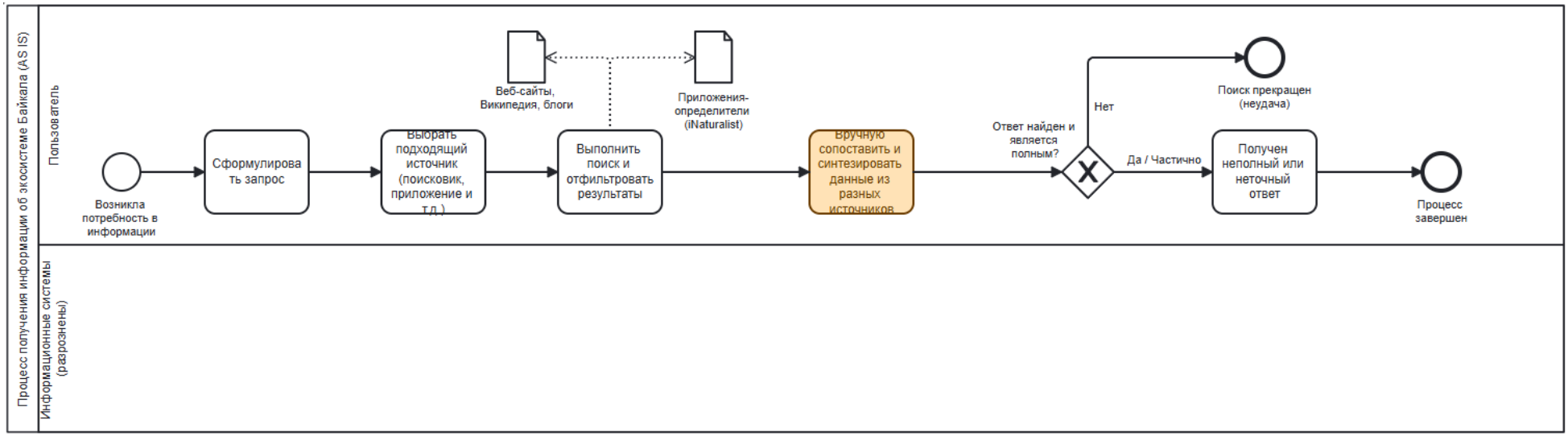

Рисунок 1 – BPMN диаграмма процесса AS IS

Таким образом, модель AS IS демонстрирует, что существующий процесс является неэффективным, требует от пользователя значительных временных затрат и экспертных навыков поиска, не гарантируя при этом получения достоверного и комплексного результата.

3 Модель процесса TO BE: Оптимизация с помощью "Эко-ассистента"

Разрабатываемый "Эко-ассистент Байкала" призван кардинально трансформировать описанный выше процесс, выступая в роли единой точки входа и автоматизированного инструмента для получения информации. Диаграмма процесса TO BE (Рисунок 2 – BPMN диаграмма процесса TO BE) моделирует новый, оптимизированный рабочий поток.

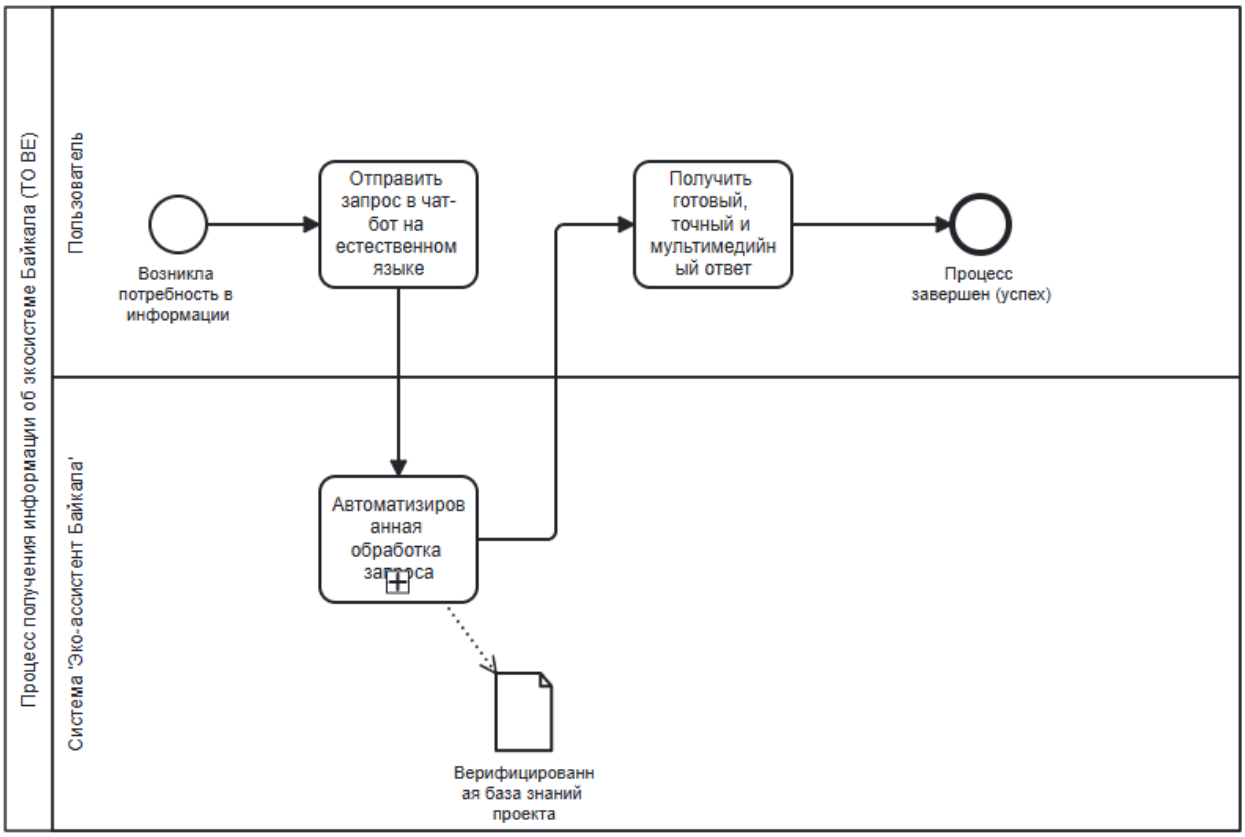

Рисунок 2 – BPMN диаграмма процесса TO BE

Ключевые изменения и преимущества нового процесса:

1. Упрощение действий пользователя: в модели TO BE единственными активными действиями пользователя являются формулировка запроса на естественном языке и получение готового ответа. Сложные этапы выбора источника, фильтрации и синтеза данных полностью исключены из его зоны ответственности.

2. Автоматизация обработки: Вся сложность процесса переносится на сторону системы. Подпроцесс "Автоматизированная обработка запроса" инкапсулирует в себе всю внутреннюю логику:

   a. Анализ запроса: Система, используя гибридную архитектуру NLU-фреймворка и большой языковой модели (LLM), точно распознает намерение и извлекает все параметры запроса.

   b. Взаимодействие с базой знаний: в отличие от поиска по всей сети Интернет, ассистент обращается к собственной специализированной и верифицированной базе данных, что гарантирует точность и достоверность информации.

с. Генерация мультимедийного ответа: Система автоматически агрегирует все необходимые компоненты (текстовое описание, изображения по заданным признакам, интерактивную карту) в единый, комплексный ответ.

3. Гарантированное завершение и определенность результата: в отличие от модели AS IS, где исход непредсказуем и часто приводит к прекращению поиска пользователем, процесс TO BE всегда завершается определенным и полезным для пользователя результатом. Даже в случае отсутствия информации в базе знаний, система не оставляет пользователя в неведении, а предоставляет четкий ответ. Это делает "Эко-ассистент" не просто чат-ботом, а полноценным и, что ключевое, надежным интерактивным инструментом для исследования региона.

4 Примеры сценариев использования в процессах AS IS и TO BE

Для иллюстрации практических различий между процессами AS IS и TO BE рассмотрим гипотетический, но репрезентативный сценарий. Предположим, пользователь (например, студент-биолог) пытается найти ответ на узкоспециализированный запрос: "Какие эндемичные мхи произрастают на скальных выходах острова Ольхон?"

Сценарий в рамках процесса AS IS:

1. Взаимодействие с поисковыми системами и общими ассистентами: при вводе данного запроса в поисковую систему или диалоге с общим ассистентом (например, Яндекс Алисой) пользователь получит набор неструктурированных ссылок. Результаты поиска будут включать общие статьи о флоре Ольхона, туристические форумы, возможно, научные публикации без прямого ответа и коммерческие предложения. Пользователь будет вынужден потратить значительное время на ручную фильтрацию и анализ источников, с высокой вероятностью, не найдя точного ответа. Результат: большие временные затраты при неопределенном исходе.

2. Взаимодействие с большими языковыми моделями (LLM) общего назначения: при обращении с тем же запросом к неспециализированной LLM (например, ChatGPT, GigaChat) существует значительный риск получения фактологически некорректного ответа, известного как "галлюцинация". Модель может сгенерировать правдоподобный, но вымышленный список видов (например, "Сфагнум байкальский") или неверно классифицировать существующие организмы. Результат: получение дезинформации, что подрывает доверие к технологии и может привести к ошибкам в дальнейшей работе пользователя.

Оба варианта в рамках процесса AS IS демонстрируют его ключевые недостатки: неэффективность и ненадежность.

Сценарий в рамках процесса ТО ВЕ (с использованием "Эко-ассистента"):

При обращении с тем же запросом к "Эко-ассистенту Байкала" система инициирует четкий и надежный рабочий поток, который приводит к одному из двух полезных для пользователя исходов:

1. Исход 1 (Данные найдены): система анализирует запрос, обращается к своей верифицированной базе знаний, находит релевантную информацию и предоставляет пользователю структурированный, мультимедийный ответ, включающий список видов, их описание и, возможно, карту с точками находок. Результат: быстрый и точный ответ.

2. Исход 2 (Данные не найдены): если в базе знаний отсутствует информация по столь специфическому запросу, система формирует честный и однозначный ответ, например: "В моей верифицированной базе знаний отсутствует информация об эндемичных мхах, произрастающих именно на скалах острова Ольхон".

Польза такого ответа заключается не в предоставлении искомых данных, а в создании информационной определенности. Пользователь экономит время, которое было бы потрачено на безрезультатный ручной поиск, и, что критически важно, защищен от дезинформации. Такой подход формирует доверие к системе как к надежному и честному источнику специализированных знаний, который четко осознает границы своей компетенции.

Процесс ТО ВЕ, реализуемый "Эко-ассистентом", трансформирует пользовательский опыт. Он заменяет неопределенность, ручной труд и риск получения ложной информации на скорость, точность и надежность. Даже в случае отсутствия данных, система предоставляет пользователю ценный результат — определенность и экономию времени, что является фундаментальным улучшением по сравнению с существующим процессом.

5 Заключение по анализу процессов

Сравнение моделей AS IS и ТО ВЕ наглядно доказывает актуальность и практическую значимость проекта. "Эко-ассистент Байкала" решает фундаментальную проблему фрагментации информации, автоматизируя рутинные и трудоемкие задачи пользователя и предоставляя ему мощный, надежный и удобный инструмент для взаимодействия с уникальной экосистемой Байкальского региона.

Описание вариантов использования

Для определения и формализации функциональных требований к системе "Эко-ассистент Байкала" была разработана диаграмма вариантов использования (Use Case Diagram) в соответствии со стандартом UML (Unified Modeling Language). Данная диаграмма описывает систему с точки зрения пользователя, отвечая на вопрос "что система должна делать?", и служит основой для дальнейшего проектирования и реализации функционала.

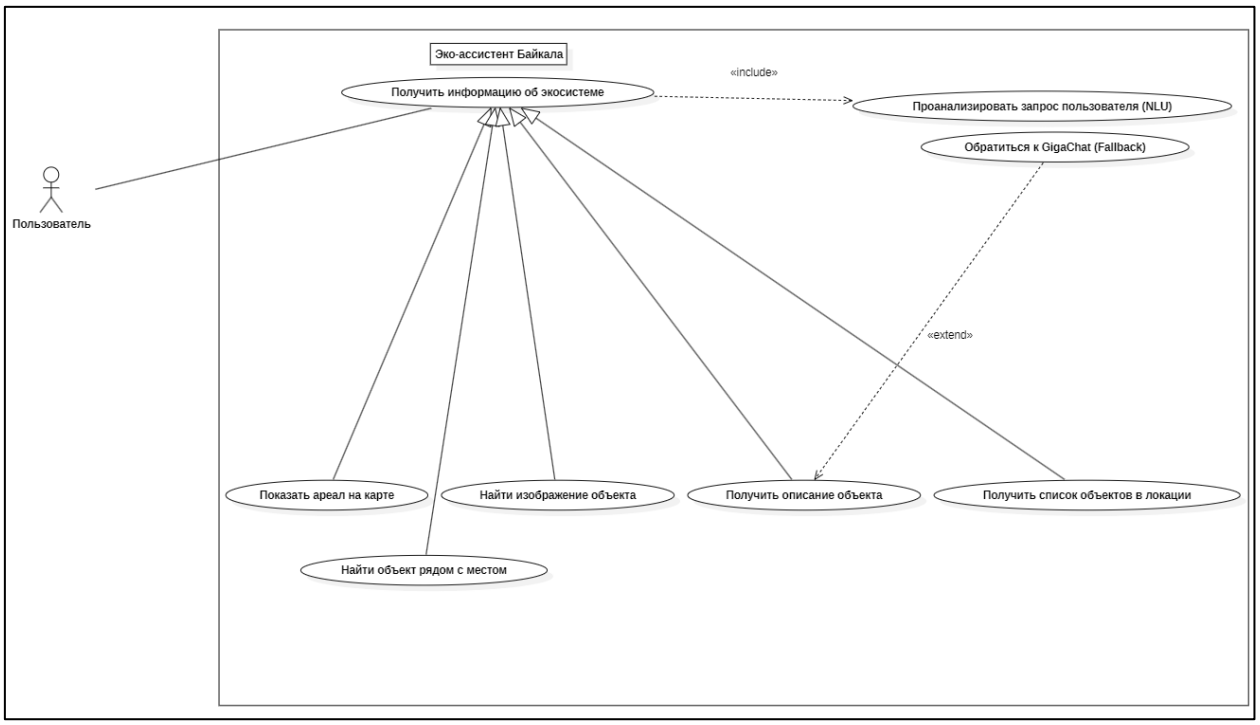

Рисунок 3 – Диаграмма вариантов использования (Use Case)

6 Описание компонентов диаграммы

Действующие лица (Actors):

"Пользователь": Единственное действующее лицо, представляющее всех конечных пользователей системы (туристов, студентов, исследователей). Пользователь инициирует все взаимодействия с системой с целью получения информации.

Граница системы (System Boundary):

"Эко-ассистент Байкала": Прямоугольник, обозначающий границы разрабатываемой системы. Все варианты использования (Use Cases), находящиеся внутри, являются функциями, реализуемыми непосредственно ассистентом.

Варианты использования (Use Cases) и их отношения:

Базовый вариант использования (Parent Use Case):

"Получить информацию об экосистеме": является обобщенным (родительским) вариантом использования, который отражает основную цель Пользователя. Он напрямую не выполняется, а конкретизируется через дочерние Use Cases. Пользователь напрямую ассоциирован с этой целью.

Специализированные варианты использования (Child Use Cases):

Эти варианты наследуются от базового, представляя собой конкретные способы получения информации:

– "Получить описание объекта": Цель пользователя – получить текстовую справку о конкретном виде флоры или фауны.

– "Найти изображение объекта": Цель – получить визуальное представление объекта, возможно, с учетом определенных признаков (например, "зимой").

– "Показать ареал на карте": Цель – увидеть на карте общую область распространения вида.

– "Найти объект рядом с местом": более сложный гео-запрос, целью которого является поиск мест обитания вида вблизи указанной локации.

– "Получить список объектов в локации": Цель – получить перечень всех известных видов, встречающихся в заданной географической области.

Обязательный служебный вариант использования (с отношением <<include>>):

– "Проанализировать запрос пользователя (NLU)": Данный Use Case является обязательной частью основного сценария "Получить информацию об экосистеме". Отношение <<include>> показывает, что для выполнения любого информационного запроса система всегда и в обязательном порядке должна сначала выполнить анализ текста пользователя для распознавания его намерения и извлечения сущностей. Это ядро интеллектуальной составляющей системы.

Опциональный служебный вариант использования (с отношением <<extend>>):

– "Обратиться к GigaChat (Fallback)": Этот Use Case расширяет функционал варианта "Получить описание объекта". Отношение <<extend>> показывает, что данное действие является опциональным и выполняется только при определенных условиях: если в основной верифицированной базе знаний не нашлось данных, и, если сам Пользователь предварительно активировал данную функцию в настройках. Такое моделирование точно отражает архитектурное решение об ограниченном и контролируемом применении внешних LLM для сохранения достоверности данных.

7 Перспективы расширения функционала

Представленная диаграмма описывает функциональные возможности системы на текущем этапе реализации. Важно отметить, что выбранная архитектура и группировка функций с помощью наследования (Generalization) обеспечивают высокую расширяемость системы. В будущем набор вариантов использования может быть легко дополнен новыми функциями. Эти новые возможности логично впишутся в существующую структуру как новые дочерние элементы общего варианта использования "Получить информацию об экосистеме", не требуя кардинального пересмотра архитектуры.

Выработка требований и постановка задачи

Настоящий раздел формализует требования к программному продукту "Эко-ассистент Байкала" и определяет комплексную задачу для его разработки. Требования были выработаны на основе анализа предметной области, изучения существующих решений, а также моделирования бизнес-процессов (AS IS и TO BE) и вариантов использования (Use Case).

8 Требования к системе

Требования к системе декомпозируются на три ключевые категории: функциональные, нефункциональные и системные.

– ФТ-1: Анализ запроса пользователя: Система должна обеспечивать анализ входящих текстовых сообщений от пользователя на естественном языке для определения его намерения и извлечения ключевых сущностей (название объекта, географическая локация, признаки);

– ФТ-2: Предоставление текстового описания: Система должна предоставлять пользователю возможность получить верифицированную текстовую информацию (описание, факты) о запрашиваемом объекте флоры или фауны;

– ФТ-3: Предоставление изображений: Система должна предоставлять пользователю галерею изображений по запросу о биологическом объекте;

– ФТ-4: Отображение ареала на карте: Система должна быть способна сгенерировать и отобразить карту с общим ареалом обитания запрашиваемого вида;

– ФТ-5: Поиск объектов вблизи локации: Система должна предоставлять пользователю возможность найти места обитания указанного вида вблизи заданной географической точки;

– ФТ-6: Формирование списков объектов: Система должна уметь формировать и предоставлять списки видов флоры и фауны, встречающихся в указанной пользователем локации;

– ФТ-7: Управляемый фоллбэк-механизм: Система должна включать опциональный механизм фоллбэка к большой языковой модели (GigaChat) для получения текстового описания в случае отсутствия данных в основной базе. Данная функция должна быть активируема пользователем в настройках;

– ФТ-8: Переключение режимов обработки: Система должна предоставлять пользователю возможность вручную переключаться между двумя режимами NLU-обработки через меню настроек.

– НФТ-1: Надежность и достоверность информации: это ключевое требование. Система должна использовать в качестве основного источника только специализированную и верифицированную базу данных. Применение внешних LLM должно быть строго ограничено для предотвращения фактологических ошибок и "галлюцинаций";

– НФТ-2: Производительность: Время ответа системы на простой запрос (например, получение описания) не должно превышать 5 секунд. Для комплексных запросов, требующих генерации карты, время ответа не должно превышать 15 секунд;

– НФТ-3: Удобство использования: Взаимодействие с системой должно происходить в интуитивно понятном диалоговом режиме через интерфейс мессенджера Telegram. Система должна корректно обрабатывать вариативные формулировки на естественном русском языке;

– НФТ-4: Расширяемость: Архитектура системы должна быть модульной и позволять в будущем добавлять новые функции (варианты использования) и источники данных без необходимости полной переработки существующих компонентов;

– НФТ-5: Доступность: Система должна быть доступна для пользователей в режиме 24/7.

– СТ-1: Система должна быть реализована в виде чат-бота для платформы Telegram и взаимодействовать с пользователями через Telegram Bot API.

– СТ-2: Система должна быть реализована на основе архитектуры с двумя независимыми, переключаемыми пользователем режимами (конвейерами) обработки запросов: один на базе NLU-фреймворка Rasa, второй — на базе большой языковой модели GigaChat.

– СТ-3: Система должна взаимодействовать с внешним бэкенд-сервисом (API) для доступа к верифицированной базе знаний и генерации мультимедийного контента.

9 Постановка задачи

На основании вышеизложенных требований, ставится следующая комплексная задача на разработку:

Разработать и реализовать программный комплекс "Эко-ассистент Байкала" в виде Telegram-бота, предназначенного для предоставления пользователям верифицированной мультимедийной информации о флоре и фауне Байкальского региона.

Для достижения поставленной цели необходимо выполнить следующие подзадачи:

– Спроектировать микросервисную архитектуру программного комплекса, включающую модуль взаимодействия с Telegram, два независимых NLU-конвейера и модуль интеграции с внешним API.

– Реализовать модуль взаимодействия с пользователем, обеспечивающий прием, отправку, корректное отображение сообщений и интерфейс для переключения режимов в Telegram.

– Реализовать два независимых конвейера обработки NLU-запросов: конвейер на основе фреймворка Rasa для распознавания структурированных команд; конвейер, использующий большую языковую модель GigaChat для семантического анализа запросов на естественном языке.

– Реализовать модуль генерации ответов, отвечающий за формирование запросов к внешнему API и преобразование полученных данных в удобный для пользователя формат (текст, галереи изображений, интерактивные карты).

– Реализовать опциональный фоллбэк-механизм для текстовых запросов с возможностью его активации пользователем.

– Провести комплексное тестирование системы, включая функциональное тестирование (проверка всех вариантов использования в обоих режимах), тестирование производительности и оценку качества пользовательского опыта (UX).