

Использование технологий ИИ в проектировании элементов компьютерных систем

А. А. Койбаш, Р. В. Мальчева

E-mail: mr.koibash@yandex.ru, raisa.malcheva@yandex.ru

Аннотация

В статье выполнен обзор основных технологий искусственного интеллекта, которые применяются при проектировании компьютерных систем. Рассмотрены перспективы их развития. В рамках исследования разработан интеллектуальный ассистент для помощи инженерам-проектировщикам. Описаны выбор основных компонентов, таких как система управления базами данных, модель ИИ, аппаратные ускорители, а также реализация прототипа ассистента. Намечены направления для будущих исследований.

Введение

Технологии искусственного интеллекта (ИИ) находят свое применение в различных областях, таких как компьютерная графика [1, 2] и многие другие. В настоящее время системы автоматизированного проектирования (САПР) также претерпевают значительные изменения благодаря интеграции технологий искусственного интеллекта. Эта тенденция обусловлена растущей сложностью проектируемых систем и необходимостью повышения эффективности процесса проектирования. Анализ современного состояния исследований в данной области показывает активное развитие различных направлений применения искусственного интеллекта: от экспертных систем до глубокого обучения. Особый интерес представляет изучение возможностей использования этих технологий для автоматизации различных этапов проектирования и улучшения качества принимаемых решений.

В России планируется создать систему ИИ для проектирования цифровых микросхем. Проект рассчитан на период с 2024 по 2026 гг. и направлен на разработку «интеллектуальной» САПР цифровых сверхбольших интегральных схем (СБИС) [3]. Проект рассчитан на период с 2024 по 2026 гг. и направлен на разработку «интеллектуальной» САПР СБИС. В основе разработки будет лежать открытый маршрут OpenLane, который уже успешно применяется в реальных проектах. Ожидается, что новая система позволит улучшить характеристики СБИС примерно на 10–20 % и повысить эффективность процесса разработки. Проект также предполагает оптимизацию энергопотребления и производительности, разработку моделей машинного обучения, поддержку средств проектирования отечественных технологических процессов и

создание центров компетенции для внедрения и технической поддержки.

Искусственный интеллект в проектировании

Сверточные нейронные сети (CNN) успешно применяются для анализа топологии печатных плат и проектирования интегральных схем [4]. Рекуррентные нейронные сети (RNN) и их современные варианты, такие как LSTM и GRU, показывают эффективность в моделировании временных последовательностей и прогнозировании поведения сложных систем [5]. Особое значение приобретают методы обучения с подкреплением (Reinforcement Learning), позволяющие системам ИИ обучаться оптимальным стратегиям проектирования через взаимодействие с виртуальной средой. Данный подход особенно перспективен для решения задач оптимизации архитектуры и параметров проектируемых систем.

Применение нейронных сетей в проектировании компьютерных систем представляет собой отдельное перспективное направление. Современные архитектуры нейронных сетей позволяют решать широкий спектр задач, от оптимизации топологии до предсказания характеристик проектируемых систем [6]. Глубокие нейронные сети демонстрируют особую эффективность в задачах:

- автоматической генерации схемотехнических решений;
- оптимизации размещения компонентов;
- прогнозирования тепловых и электрических характеристик;
- анализа надежности проектируемых систем.

Особый интерес представляют генеративно-состязательные сети (GAN), способные создавать новые варианты проектных решений на основе обучающих данных. Такие

сети могут генерировать множество альтернативных вариантов топологии или архитектуры системы, удовлетворяющих заданным ограничениям.

Появление больших языковых моделей (Large Language Models, LLM) открыло новые возможности в области автоматизированного проектирования. В отличие от традиционных систем автоматизации, LLM способны понимать и генерировать естественные языковые описания технических решений, что существенно упрощает взаимодействие между проектировщиком и системой [7].

Современные языковые модели, такие как GPT, BERT и их производные, демонстрируют способность к пониманию контекста и генерации осмысленных рекомендаций по проектированию [8]. Особенно важным является их способность к обобщению накопленных знаний и применению их в новых контекстах. Это позволяет использовать LLM для:

- анализа технических требований и спецификаций;
- генерации вариантов проектных решений;
- документирования процесса проектирования;
- верификации соответствия проектных решений требованиям;
- оптимизации существующих проектов.

Важной особенностью применения LLM в проектировании является возможность их интеграции с существующими САПР через специализированные программные интерфейсы. Это позволяет создавать гибридные системы, сочетающие преимущества традиционных инструментов автоматизации с возможностями искусственного интеллекта.

Применение промпт-инжиниринга.

Промпт-инжиниринг представляет собой новое направление в области применения языковых моделей, особенно актуальное для задач проектирования компьютерных систем. Данный подход заключается в разработке специализированных запросов (промптов), позволяющих получить от языковой модели максимально релевантные и точные результаты.

В контексте проектирования элементов компьютерных систем промпт-инжиниринг включает несколько ключевых аспектов. Прежде всего, это разработка структурированных шаблонов запросов, учитывающих специфику предметной области. Такие шаблоны должны содержать всю необходимую контекстную информацию и четкие инструкции для языковой модели.

Особое значение приобретает итеративный подход к формированию промптов, когда каждый последующий запрос учитывает результаты предыдущих взаимодействий с

моделью. Это позволяет постепенно уточнять и оптимизировать проектные решения. Например, при проектировании интерфейса системы промпты могут последовательно уточнять требования к производительности, энергопотреблению и физическим параметрам компонентов.

Важным аспектом является также разработка системы валидации получаемых результатов. Это может включать как автоматическую проверку на соответствие техническим ограничениям, так и механизмы экспертной оценки генерируемых решений.

Интеграция технологий ИИ в процесс проектирования

Эффективное применение технологий искусственного интеллекта в проектировании требует их глубокой интеграции с существующими процессами и инструментами. Важным аспектом является разработка методологии, позволяющей органично сочетать традиционные подходы с возможностями ИИ.

Современные подходы к такой интеграции предполагают создание многоуровневых систем, где различные технологии ИИ применяются на разных этапах проектирования [9]. На концептуальном уровне могут использоваться языковые модели для анализа требований и генерации архитектурных решений. На уровне детального проектирования применяются специализированные нейронные сети для оптимизации конкретных параметров и характеристик. Особое внимание уделяется созданию интерфейсов взаимодействия между различными компонентами системы проектирования. Это включает разработку стандартизированных форматов данных, протоколов обмена информацией и механизмов синхронизации различных инструментов проектирования [10, 11].

Важным аспектом является также обеспечение прослеживаемости и объяснимости решений, принимаемых системами ИИ. Это необходимо как для валидации проектных решений, так и для соответствия нормативным требованиям в области проектирования критически важных систем.

Перспективы развития технологий ИИ в проектировании КС

Дальнейшее развитие технологий искусственного интеллекта в проектировании компьютерных систем связано с несколькими ключевыми направлениями. Прежде всего, это совершенствование существующих моделей и алгоритмов, повышение их точности и эффективности. Особое внимание уделяется развитию методов, позволяющих работать с

ограниченными наборами данных, что особенно актуально для специализированных областей проектирования [9].

Одним из наиболее перспективных направлений является развитие интеллектуальных ассистентов, способных предоставлять экспертную поддержку в процессе проектирования. Такие системы, основанные на современных языковых моделях, могут не только давать рекомендации по техническим решениям, но и обучаться на основе накопленного опыта взаимодействия с инженерами-проектировщиками, постоянно повышая качество своих рекомендаций.

Перспективным направлением является также разработка гибридных систем, сочетающих различные подходы к искусственному интеллекту. Интеграция интеллектуальных ассистентов с традиционными САПР и специализированными нейронными сетями может обеспечить более полное покрытие задач проектирования и повысить надежность принимаемых решений.

Развитие технологий федеративного обучения без необходимости обмена данных между клиентскими устройствами и сервером может способствовать созданию распределенных систем проектирования, где различные организации могут совместно использовать накопленный опыт без прямого обмена конфиденциальными данными. Это особенно важно для развития интеллектуальных ассистентов, которые могут обогащать свою базу знаний, сохраняя при этом конфиденциальность проектных решений отдельных организаций.

Разработка интеллектуального ассистента

В рамках исследования разработан интеллектуальный ассистент для помощи инженерам. При создании прототипа определены и реализованы следующие ключевые требования: интуитивно понятное пользовательское взаимодействие, максимальное использование потенциала современных больших языковых моделей, а также широкий потенциал применения – возможность использования ассистента как в связке с существующими системами автоматизированного проектирования, так и в качестве самостоятельного инструмента консультационной поддержки.

Архитектура предлагаемого решения

В рамках исследования разработан интеллектуальный ассистент системы автоматизированного проектирования, реализованный в виде программного комплекса с использованием API мессенджера Telegram.

Система построена на принципах объектно-ориентированного программирования.

Основой системы является модульный принцип организации программного кода. Использована классическая «слоеная» архитектура, в основе которой лежит принцип различной ответственности для каждого из участков программы. Схема такой архитектуры изображена на рисунке 1.

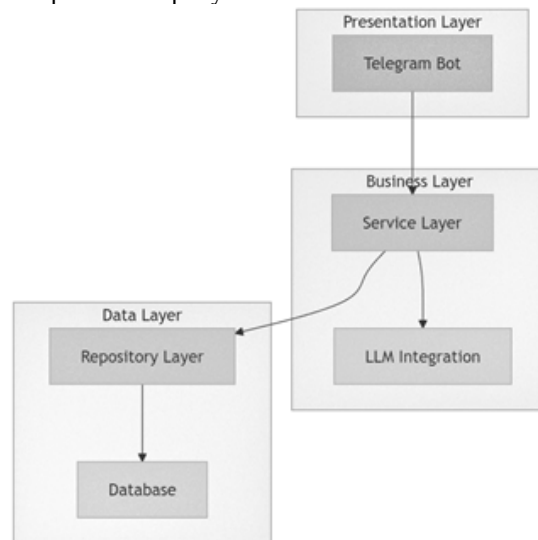


Рисунок 1 – Архитектура приложения

Центральным элементом выступает модуль Telegram Bot, обеспечивающий взаимодействие с пользователем. Он реализованный через API Telegram и является слоем презентации (точкой входа). Данный модуль обеспечивает прием пользовательских запросов и отправку ответов системы.

Второй ключевой компонент – модуль обработки естественного языка, построенный на базе современной языковой модели. Этот модуль отвечает за анализ пользовательских запросов и генерацию релевантных ответов в контексте автоматизированного проектирования.

Система хранения данных, основанная на реляционной базе данных, обеспечивает долговременное сохранение и восстановление пользовательских сессий (рисунок 2).

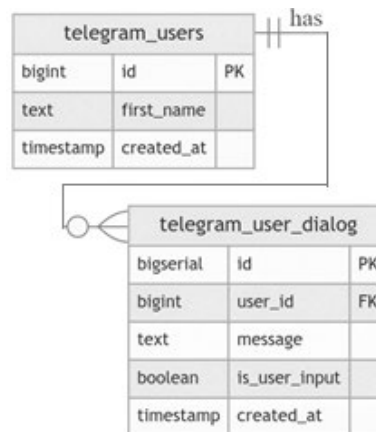


Рисунок 2 – Таблицы базы данных

В рамках прототипа необходимо и достаточно наличие двух таблиц: пользователей (`telegram_users`) и пользовательских диалогов (`telegram_user_dialog`) с целью обеспечения возможности учёта контекста предыдущих сообщений интеллектуальным ассистентом.

В рамках программного комплекса используется паттерн `Service Locator`, который предоставляет единую точку доступа ко всем зависимостям. Благодаря этому, любой класс имеет возможность получить доступ к централизованному реестру всех сервисов приложения. Данный подход обеспечивает гибкость конфигурации и масштабируемость решения.

Также для приложения разработана подсистема логирования и мониторинга, позволяющая отслеживать работу всех компонентов системы и обеспечивающая возможность анализа эффективности её работы.

Выбор технологий

При разработке системы особое внимание было уделено выбору технологического стека. В качестве основного языка программирования выбран Python, что обусловлено его широкими возможностями в области обработки естественного языка и машинного обучения, а также развитой экосистемой библиотек для разработки программных систем.

Для работы с языковой моделью применяется библиотека `llama.cpp`, обеспечивающая эффективное использование вычислительных ресурсов и гибкую настройку параметров модели. Данный выбор обоснован возможностью оптимизации работы модели под различные устройства и поддержкой широкого спектра форматов взаимодействия.

Взаимодействие с пользователем организовано через программный интерфейс Telegram посредством библиотеки `telebot`, что обеспечивает надежную и масштабируемую платформу для обмена сообщениями. Данное решение позволяет абстрагироваться от низкоуровневых особенностей сетевого взаимодействия и сосредоточиться на логике работы системы.

Разработка системы промптов

В рамках исследования разработана специализированная система промптов, учитывающая специфику задач автоматизированного проектирования. Базовый системный промпт определяет роль виртуального ассистента и задает основные параметры его поведения в контексте решения проектных задач.

Особое внимание уделено механизму поддержания контекста диалога, что достигается путем сохранения и анализа истории взаимодействия с пользователем. Система промптов включает также временные метки и

контекстную информацию, что позволяет генерировать более релевантные и актуальные ответы.

Посредством использования системного промпта языковой модели заданы основные директивы: консультирование по техническим вопросам, анализ проектных решений, предоставление рекомендаций, помощь в поиске ошибок. К системному промпту в ходе каждого взаимодействия добавляется текущее время, чтобы ассистент мог учитывать этот параметр в ходе диалога.

Реализация прототипа

Для наглядной демонстрации работы интеллектуального ассистента создан прототип программного продукта на локальном компьютере. В процессе реализации были заложены удобные архитектурные паттерны, дающие возможность дальнейшего расширения. Система протестирована на условиях, максимально приближенных к реальным.

Описание программной реализации

Программная реализация системы выполнена в соответствии с принципами объектно-ориентированного программирования и современными практиками разработки программного обеспечения. Диаграмма классов приложения изображена на рисунке 3. Центральным элементом реализации является класс `TelegramBot`, включающий в себя использование функционала взаимодействия с Telegram API. Данный класс обеспечивает основную функциональность системы и координирует работу остальных компонентов.

Взаимодействие с языковой моделью реализовано через специализированный класс `Llm`, инкапсулирующий логику работы с моделью и предоставляющий высокоуровневый интерфейс для генерации ответов. Конфигурация модели осуществляется через внешний конфигурационный файл, что обеспечивает гибкость настройки параметров без необходимости модификации программного кода. В системе реализован механизм логирования всех взаимодействий, что позволяет отслеживать работу системы и анализировать качество генерируемых ответов. Данный механизм является критически важным для дальнейшего совершенствования системы и адаптации её работы под конкретные задачи проектирования.

В качестве системы управления базами данных (СУБД) выбрана PostgreSQL, что обусловлено рядом существенных технических и эксплуатационных преимуществ данной СУБД. PostgreSQL представляет собой объектно-реляционную систему управления базами данных. Ключевым фактором выбора PostgreSQL является её архитектурная надёжность,

обеспечивающая высокий уровень целостности данных благодаря встроенным механизмам репликации и продвинутым системам резервного копирования. Существенное значение имеет также и производительность системы, достигаемая за счёт эффективной обработки конкурентных запросов. В контексте разрабатываемой системы особую значимость имеет возможность интеграции PostgreSQL с

различными языками программирования и платформами, что обеспечивает необходимую гибкость при реализации программных компонентов. СУБД развёрнута на удалённом сервере с операционной системой Ubuntu 22.04. Подключение доступно по внешнему IP-адресу для любого вычислительного устройства. Схема базы данных реализована согласно выбранной структуре в виде двух таблиц (рисунок 4).

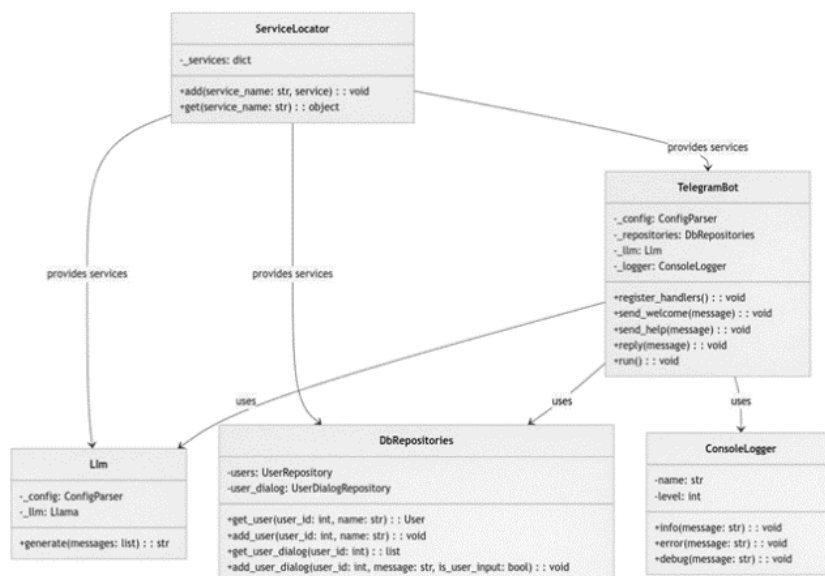


Рисунок 3 – Диаграмма классов (основная архитектура приложения)

```

a1cad=# \dt
Schema | List of relations | Type | Owner
public | telegram_user_dialog | table | a1cad_dba
public | telegram_users | table | a1cad_dba
(2 rows)

a1cad=# \d telegram_users;
Table "public.telegram_users"
Column | Type | Collation | Nullable | Default
id | bigint | | not null |
first_name | text | | not null |
created_at | timestamp without time zone | | not null | CURRENT_TIMESTAMP
Indexes:
"telegram_users_pkey" PRIMARY KEY, btree (id)
Referenced by:
TABLE "telegram_user_dialog" CONSTRAINT "telegram_user_dialog_user_id_fkey" FOREIGN KEY (user_id) REFERENCES telegram_users(id) ON UPDATE CASCADE ON DELETE CASCADE

a1cad=# \d telegram_user_dialog;
Table "public.telegram_user_dialog"
Column | Type | Collation | Nullable | Default
id | bigint | | not null | nextval('telegram_user_dialog_id_seq'::regclass)
user_id | bigint | | not null |
message | text | | not null |
is_user_input | boolean | | not null |
created_at | timestamp without time zone | | not null | CURRENT_TIMESTAMP
Indexes:
"telegram_user_dialog_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
"telegram_user_dialog_user_id_fkey" FOREIGN KEY (user_id) REFERENCES telegram_users(id) ON UPDATE CASCADE ON DELETE CASCADE
    
```

Рисунок 4 – Диаграмма классов (основная архитектура приложения)

Выбор модели ИИ

В качестве большой языковой модели выбран Qwen 2.5 с 7 миллиардами параметров. Данная нейронная сеть разработана исследовательской командой компании Alibaba и представляет собой модель с открытым исходным кодом. Такой выбор был сделан по нескольким причинам. Прежде всего, модель демонстрирует впечатляющие способности в работе с русским языком, что критически важно для русскоязычных пользователей САПР-систем.

В отличие от многих других конкурентов с открытым исходным кодом, показывающих посредственные результаты при работе с русским языком, Qwen 2.5 обеспечивает естественное и грамотное общение, включая корректное использование технической терминологии в области проектирования и инженерных расчётов.

Другим существенным преимуществом является компактный размер модели – 7 миллиардов параметров представляют собой разумный компромисс между

производительностью и требованиями к вычислительным ресурсам. Это позволяет развернуть модель на относительно доступном оборудовании, сохраняя при этом высокое качество ответов по техническим вопросам. Кроме того, Qwen 2.5 7B обладает хорошими показателями в работе с контекстом и демонстрирует способность удерживать в памяти детали предыдущих взаимодействий, что особенно важно при обсуждении сложных технических задач и многоэтапных процессов проектирования.

Для разработки прототипа использована квантизованная версия данной модели. Её размер составляет 3.3 гигабайта, что позволяет эффективно использовать аппаратную акселерацию вычислений.

Аппаратное ускорение

В целях оптимизации производительности системы и минимизации времени отклика при обработке запросов задействован графический ускоритель (GPU) в качестве специализированного аппаратного акселератора для выполнения матричных вычислений нейронной сети. Данное архитектурное решение обосновано тем, что современные графические процессоры обладают значительным преимуществом перед центральными процессорами (CPU) в задачах параллельной обработки данных, характерных для работы нейронных сетей.

В рамках реализации системы библиотека llama.cpp была скомпилирована с поддержкой технологии CUDA (Compute Unified Device Architecture) от компании NVIDIA. CUDA представляет собой программно-аппаратную архитектуру параллельных вычислений, которая позволяет существенно увеличить вычислительную производительность за счет использования графических процессоров NVIDIA. Данная технология обеспечивает прямой доступ к набору инструкций графического процессора и управление его памятью, что позволяет добиться эффективной параллельности вычислительных операций.

Интеграция компонентов

Интеграция компонентов системы осуществляется через механизм инверсии зависимостей, что обеспечивает слабую связанность модулей и возможность их независимого тестирования и модификации. Взаимодействие между компонентами организовано через четко определенные интерфейсы, что минимизирует влияние изменений в одном модуле на работу других частей системы.

Особое внимание уделено организации хранения данных. Реализованная система репозитория обеспечивает унифицированный доступ к данным пользователей и истории

взаимодействий. Такой подход позволяет абстрагироваться от конкретной реализации хранилища данных и обеспечивает возможность масштабирования системы.

Тестирование системы

В ходе тестовой эксплуатации система продемонстрировала способность эффективно решать различные задачи в области автоматизированного проектирования.

Типичный сценарий использования включает следующие этапы взаимодействия:

- инициацию диалога;
- уточнение требований;
- генерацию проектного решения;
- обсуждение альтернатив и детализацию выбранного варианта.

При работе с запросами система демонстрирует способность к контекстному анализу и генерации структурированных ответов, учитывающих специфику предметной области. Ответы системы включают не только конкретные технические решения, но и обоснование принятых решений, что особенно важно в контексте проектирования технических систем.

Тестирование разработанной системы проводилось в несколько этапов. На первом этапе осуществлялось модульное тестирование отдельных компонентов, что позволило верифицировать корректность их работы в изоляции. Далее проводилось интеграционное тестирование, направленное на проверку корректности взаимодействия между компонентами системы.

Особое внимание было уделено тестированию качества генерируемых ответов. Для этого был разработан тестовые сценарии, охватывающих различные аспекты проектирования компьютерных систем. Результаты тестирования показали высокую релевантность генерируемых ответов и способность системы адаптироваться к различным типам запросов.

Выводы

В данной статье рассмотрены основные технологии ИИ, которые используют исследователи и практики при проектировании элементов компьютерных систем.

В качестве примера описана разработка ассистента инженера-проектировщика, включающая выбор основных компонентов, таких как система управления базами данных, модель ИИ, аппаратные ускорители, а также реализацию прототипа ассистента.

Направлением дальнейших исследований является разработка САПР для демонстрации применения технологий ИИ в учебном процессе при подготовке инженеров-проектировщиков компьютерных систем.

Литература

1. Мулявин, Д. Е. Расширение возможностей систем генерации изображений путем использования нейронных сетей / Д. Е. Мулявин, Р. В. Мальчева, А. А. Койбаш // Информатика и кибернетика. – Донецк: ДонНТУ, 2024. - № 3(37). – С. 13-18.
2. Волгушева, А. И. Применение сверточных нейронных сетей для распознавания объектов на изображении. / А. И. Волгушева, Р. В. Мальчева // Информатика и кибернетика. – Донецк: ДонНТУ, 2024. - № 3(37). – С. 32-38.
3. В России учат ИИ проектировать цифровые микросхемы [Электронный ресурс] – UML: https://www.cnews.ru/news/top/2024-07-15_uchenyie_iz_rossii_nauchat_iskusstvennyj
4. Peter, S. M. Lithographic Hotspot Detection Using CNN / S. M. Peter, K. L. Nisha, M. S. Arun Sankar // IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2024. - C. 1-5.
5. Faraji, A. A Hybrid Approach Based on Recurrent Neural Network for Macromodeling of Nonlinear Electronic Circuits / A. Faraji, S. A. Sadrossadat, M. Yazdian-Dehkordi, M. Nabavi, Y. Savaria // IEEE Access, 2022. - №10. - C. 127996-128006.
6. Савостин, Д. А. Будущие перспективы автоматизированного проектирования (САПР) с точки зрения искусственного интеллекта и 3D-печати / Д. А. Савостин, Е. О. Кириченко, А. О. Шаранов // Известия ТулГУ. Технические науки, 2023.- №2.
7. Балашев, А. В. Сравнительный анализ эффективности различных моделей машинного обучения в задачах генерации контента / А. В. Балашев, М. В. Ступина // Молодой исследователь Дона, 2024. - №3.
8. Gupta, P. Generative AI: A systematic review using topic modelling techniques / P. Gupta, B. Ding, C. Guan, D. Ding // Data and Information Management, 2024. - №2. - 100066.
9. Luukko M. UTILIZATION OF ARTIFICIAL INTELLIGENCE IN ELECTRONICS DESIGN. – 2024.
10. Li, Y. Large Language Models for Manufacturing / Y. Li, H. Zhao, H. Jiang, Y. Pan, Z. Liu and others // arXiv:2410.21418 [cs.AI]. 2024. Available at: <https://arxiv.org/abs/2410.21418>.
11. Chen, G. Intelligent OPC Engineer Assistant for Semiconductor Manufacturing / G. Chen, H. Y. Yang, H. Ren, B. Yu // arXiv:2408.12775v1 [cs.AI]. 2024. Available at: <https://arxiv.org/abs/2408.12775>.

Койбаш А. А., Мальчева Р. В. Использование технологий ИИ в проектировании элементов компьютерных систем. В статье выполнен обзор основных технологий искусственного интеллекта, которые применяются при проектировании компьютерных систем. Рассмотрены перспективы их развития. В рамках исследования разработан интеллектуальный ассистент для помощи инженерам-проектировщикам. Описаны выбор основных компонентов, таких как система управления базами данных, модель ИИ, аппаратные ускорители, а также реализация прототипа ассистента. Намечены направления для будущих исследований.

Ключевые слова: САПР, искусственный интеллект, ассистент, прототип, тестирование

Koibash A. A., Malcheva R. V. The use of AI technologies in the design of computer system components. The article provides an overview of the main AI technologies used in the design of computer systems. The prospects for their development are discussed. As part of the research, an intelligent assistant has been developed to assist design engineers. The selection of key components, such as a database management system, an AI model, and hardware accelerators, as well as the implementation of the assistant prototype, are described. Directions for future research are outlined.

Keywords: CAD, AI, assistant, prototype, testing

Статья поступила в редакцию 08.06.2025
Рекомендована к публикации профессором Зори С. А.