

На правах рукописи



Видьманов Дмитрий Александрович

**МОДЕЛИ, АЛГОРИТМЫ И ПРОГРАММНЫЕ СРЕДСТВА
АДАПТИВНОГО ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА
МОБИЛЬНЫХ УСТРОЙСТВ**

2.3.5. Математическое и программное обеспечение вычислительных систем,
комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Москва – 2026

Работа выполнена в Федеральном государственном автономном образовательном учреждении высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана).

Научный руководитель: **АЛФИМЦЕВ Александр Николаевич**
доктор технических наук, доцент, заведующий
кафедрой «Информационные системы и
телекоммуникации»

Официальные оппоненты: **ХАЧУМОВ Вячеслав Михайлович**
доктор технических наук, профессор,
Федеральное государственное учреждение
«Федеральный исследовательский центр
«Информатика и управление» Российской
академии наук», главный научный сотрудник

ЧУВИЛИН Кирилл Владимирович
кандидат технических наук, доцент,
Общество с ограниченной ответственностью
«Открытая мобильная платформа», заместитель
технического директора – директор департамента

Ведущая организация: Федеральное государственное бюджетное
учреждение науки Институт проблем управления
имени В.А. Трапезникова Российской академии
наук

Защита состоится «10» декабря 2026 г. в 15:00 часов на заседании
совета по защите докторских и кандидатских диссертаций 24.2.331.19 при
МГТУ им. Н.Э. Баумана по адресу: 105005, г. Москва, ул. 2-я Бауманская, д.
5, стр. 1, зал Ученого совета ГУК МГТУ им. Н.Э. Баумана.

С диссертацией можно ознакомиться в библиотеке МГТУ им. Н.Э.
Баумана и на сайте bmstu.ru.

Отзывы на автореферат, заверенные печатью учреждения, просьба
направлять по адресу: 105005, г. Москва, ул. 2-я Бауманская, д. 5, стр. 1,
Ученому секретарю диссертационного совета 24.2.331.19, кафедра ИУ-3.

Автореферат разослан «__» _____ 2026 г.

Ученый секретарь
диссертационного совета 24.2.331.19
кандидат технических наук, доцент

С.А. Сакулин

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования. В настоящее время исследования цифровых мобильных экосистем и приложений с целью разработки интеллектуальных и адаптивных интерфейсов становится передним краем в развитии современного программного обеспечения пользовательских интерфейсов (ПИ) и элементов интерфейса (ЭИ) смартфонов, планшетов, терминалов оплаты и других мобильных устройств (МУ). Актуальность исследований в области адаптивных пользовательских интерфейсов (АПИ) обусловлена несколькими ключевыми тенденциями развития МУ и технологий:

Во-первых, стремительное развитие искусственного интеллекта (ИИ) и машинного обучения приводит к созданию интерфейсов, которые подстраиваются в процессе использования под индивидуальные потребности пользователя в режиме реального времени. Это включает предиктивные интерфейсы, интеллектуальные системы навигации и персонализированные рекомендации, что существенно улучшает пользовательский опыт.

Во-вторых, рост требований к доступности и возможностям мобильных ПИ становится обязательным стандартом, учитывая потребности пользователей с ограниченными возможностями.

В-третьих, технический прогресс МУ (например, складные экраны и биометрия) задает новые стандарты человеко-машинного взаимодействия.

Таким образом, современные исследования фокусируются на интеграции ИИ для адаптивности ПИ, обеспечении их доступности и персонализации, что делает тему актуальной и перспективной для дальнейших разработок и внедрений в мобильных приложениях и операционных системах (ОС). Актуальность исследования адаптивных систем подтверждается тем фактом, что в национальной стратегии развития искусственного интеллекта сильный ИИ определяется как тип искусственного интеллекта, который способен выполнять различные задачи, взаимодействовать с человеком и самостоятельно (без участия человека) адаптироваться к изменяющимся условиям.

Тема адаптивности систем представлена в работах McKinley P., Brun Y., Del Toro R., подтверждающих значимость выбранного направления. При этом модели адаптации в ансамблях различных систем рассматривались авторами Горбань А.Н. и Покидышева Л.И., а работы Битюцкой Е.В. и Базарова Т.Ю. определяют реакции пользователей на изменения в интерфейсах.

К настоящему времени ряд прикладных технологий проектирования интерактивных систем и пользовательских интерфейсов, а также новые направления, связанные с интеллектуальной адаптивностью, нашли отражение в работах Croft W., Raskin J., Nielsen J., Young S. Тем не менее, ключевая задача – создание АПИ МУ, не решена. На эту тему написано много цитируемых статей, в том числе о мобильных приложениях; среди них – работы Vanderdonckt J., O'Hare G., Ahmad N., Mohapatra P., Xiong H.

Одновременно с этим широкое распространение получила прикладная область машинного обучения с подкреплением, представленная в работах ученых Sutton R., Barto A., Foerster J., Albrecht S., в которых один или несколько

агентов имеют возможность восприятия состояния среды и воздействия на нее с помощью действий. Однако общего подхода применения технологий обучения с подкреплением к разработке программных средств адаптивных пользовательских интерфейсов мобильных устройств не создано.

На основании изложенного можно сделать заключение об актуальности разработки моделей, алгоритмов и программных средств АПИ МУ, включая практическую реализацию и апробацию на реальных устройствах.

Объектом исследования являются методы и программные системы пользовательских интерфейсов мобильных устройств.

Предметом исследования являются модели и алгоритмы адаптации мобильного ПИ; алгоритмы мультиагентного обучения с подкреплением для АПИ, методика оценки качества ПИ; их взаимосвязь, реализация и экспериментальное апробирование.

Целью работы является разработка и исследование моделей, алгоритмов и программных средств адаптивного пользовательского интерфейса, повышающих эффективность и удобство использования мобильных устройств. Для достижения этой цели были поставлены и решены следующие **задачи**:

1. Выполнить обзор моделей, алгоритмов и программных средств адаптивных пользовательских интерфейсов мобильных устройств на основе принципов квантификации и метрик удобства использования интерфейса.

2. Разработать алгоритм пространственной адаптации мобильного ПИ на основе мультиагентного обучения с подкреплением и комбинированной модели награды.

3. Разработать алгоритм образной адаптации элементов мобильного ПИ с использованием локальной генеративной модели.

4. Разработать алгоритм функциональной адаптации элементов мобильного ПИ с использованием модели сверточной нейронной сети.

5. Разработать программную архитектуру адаптивного ПИ мобильной ОС, основанную на сервисе межпроцессного взаимодействия.

Область исследования соответствует п.п. 4, 6, 7 паспорта специальности 2.3.5 в части проектирования и реализации интеллектуальных систем машинного обучения (п.4), исследования и разработки программных архитектур мобильных операционных систем (п.6), разработки программных средств человеко-машинных интерфейсов (п.7).

Методы исследования. В работе использовались методы, основанные на теории искусственного интеллекта, моделях квантификации пользовательского интерфейса, математическом моделировании, технологиях разработки программного обеспечения, объектно-ориентированном и нейросетевом программировании, машинном обучении с подкреплением, кластерном анализе, однофакторном дисперсионном анализе. Экспериментальные исследования выполнялись с использованием языков программирования Python, C++; библиотек Kivy, PyTorch, TensorFlow, Keras, Sklearn, NumPy, Eigen; и инструмента сборки пакетов приложений Buildozer.

Научная новизна полученных результатов:

1. Разработан алгоритм пространственной адаптации мобильного ПИ, отличающийся использованием мультиагентного обучения с подкреплением и комбинированной модели награды, дающими возможности адаптации ПИ в реальном времени с учетом пользовательских предпочтений и удобства использования.

2. Разработан алгоритм образной адаптации элементов мобильного ПИ, отличающийся использованием локальных генеративных моделей для формирования образа ЭИ, дающими возможности синтеза визуального представления ЭИ в зависимости от контекста решаемой задачи.

3. Разработан алгоритм функциональной адаптации ЭИ, отличающийся использованием модели сверточной нейронной сети для определения категории мобильного приложения, дающая возможности анализа мобильных приложений для извлечения доступных функций на основе изображения ЭИ.

4. Разработана программная архитектура адаптивного ПИ мобильной ОС, отличающаяся использованием сервиса межпроцессного взаимодействия, дающий возможности майнинга сервисов мобильной ОС, а также выбор и запуск функций различных приложений.

Теоретическая значимость. Разработаны новые алгоритмы адаптации ПИ МУ, основанные на избирательном применении современных технологий ИИ по критериям мобильной вычислимости, адаптивности в реальном времени и удобства использования, которые увеличивают возможности пользователя в отношении результативности, эффективности и удовлетворенности при работе с мобильными приложениями.

Практическая значимость диссертации подтверждается разработанными программными средствами, реализующими алгоритмы адаптации мобильного ПИ, что позволяет использовать их в различных мобильных экосистемах на основе сенсорных экранов, в том числе в операционной системе Android и отечественной мобильной ОС Аврора. На созданное программное обеспечение получены два свидетельства (№2023683047, №2022663865) Российской Федерации о государственной регистрации программ для ЭВМ.

Программный модуль адаптации мобильного ПИ прошел экспериментальную проверку на устройствах, использующих отечественную мобильную ОС Аврора. Исследования по теме диссертационной работы проводились при поддержке НИОКР № А2BAU-1125, НИР ПРИОР/СН/НУ/22/СП2/1 проекта Bauman Deep Analytics программы «Приоритет-2030». Материалы диссертационной работы используются в учебном процессе кафедры «Информационные системы и телекоммуникации» МГТУ им. Н.Э. Баумана в рамках курсов «Разработка мультисервисных приложений для мобильных ОС», «Мультимедиа технологии».

Положения, выносимые на защиту. Соискателем лично получены следующие основные научные результаты, выносимые на защиту:

1. Алгоритм пространственной адаптации мобильного ПИ, основанный на мультиагентном обучении с подкреплением и комбинированной модели награды.

2. Алгоритм образной адаптации элементов мобильного ПИ, основанный на локальных генеративных моделях.

3. Алгоритм функциональной адаптации элементов мобильного ПИ, основанный на модели сверточной нейронной сети.

4. Программная архитектура адаптивного ПИ мобильной ОС, основанная на сервисе межпроцессного взаимодействия.

Достоверность и обоснованность полученных в диссертации результатов обеспечивается всесторонним анализом научных работ российских и зарубежных исследователей; согласованностью проведенных аналитических оценок с эмпирическими данными; соответствием основных теоретических выводов полученным практическим результатам на основе разработанных алгоритмов АПИ МУ.

Апробация работы. Основные результаты диссертационной работы представлены и обсуждены на: 1-й Национальной научно-практической конференции «Фундаментальные, поисковые, прикладные исследования и инновационные проекты» (Москва, 2022); 5-м Международном форуме «Интеллектуальные системы 4-й промышленной революции» (Томск, 2023); 15-й Всероссийской научно-технической конференции «Робототехника и искусственный интеллект» (Москва, 2023); 5-й, 6-й, 7-й Международной молодежной конференции по радиоэлектронике, электротехнике и энергетике «Radio Electronics, Electrical and Power Engineering (REEPE)» (Москва, 2023–2025); Международной конференции по стратегии и инновациям «Strategy and Technology Workshop 2025 Lake Baikal Summit» компании «Huawei» (Санкт-Петербург, 2025); Международной конференции по интегрированному интеллекту и когнитивной инженерии «Integrated Intelligence and Cognitive Engineering (ICIICE)» (Dubai, United Arab Emirates, 2026).

Личный вклад соискателя состоит в исследовании математических моделей и алгоритмов АПИ МУ на основе метрик удобства использования и мультиагентного обучения с подкреплением; создании алгоритмов пространственной, образной и функциональной адаптации элементов мобильного ПИ; разработке программных средств, архитектуры, машинного обучения и оценки эффективности модуля АПИ.

Публикации. Основные результаты диссертационного исследования опубликованы в 15 научных работах, из них: 3 научные статьи в научных журналах, рекомендованных ВАК РФ; 6 научных статей, входящих в международные базы Scopus и Web of Science; 4 – в научных журналах, материалах и трудах конференций; 2 свидетельства о государственной регистрации программ для ЭВМ.

Структура и объем работы. Диссертация состоит из введения, пяти глав, заключения, списка сокращений и условных обозначений, списка литературы, двух приложений. Общий объем диссертации составляет 190 страниц текста, в том числе 62 рисунка и 15 таблиц. Список литературы включает 205 библиографических источников.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертационного исследования, сформулированы цель и задачи диссертации, научная новизна, дана общая характеристика работы, раскрыта практическая значимость, перечислены положения, выносимые на защиту, приведена структура диссертации.

В первой главе вводится определение адаптивного пользовательского интерфейса, выполняется сравнительный анализ современных математических моделей и программных средств, реализующих адаптацию мобильного ПИ.

Оценки качества интерфейсов основаны на метриках удобства использования (юзабилити), которые стали ключевыми критериями для интерактивных программных систем, охватывающих как традиционные графические интерфейсы, так и веб-приложения, а также интерактивные мобильные сервисы и приложения. Систематизированы различные формализованные метрики удобства использования для оценки качества пользовательского интерфейса (Таблица 1).

Таблица 1 – Формализованные метрики удобства использования

<p><u>Размер целевого элемента</u> (Target Element Size):</p> $TeS = \frac{1}{n} \sum_{i=1}^n a_i$ <p>где a_i возвращает 1, если площадь элемента i больше или равна площади $S \times S$ для $S = 44$ пт. (где 1 пт – типографский пункт, 1/72 дюйма или 0,3528 мм), в противном случае возвращается 0, а n – количество ЭИ.</p>
<p><u>Читабельность элементов</u> (Element Readability):</p> $TR(i_l) = \frac{\delta}{1+B(i_l)}$ <p>где $B(i_l)$ – активация элемента i в точке l, а δ – константа внимательного изучения ЭИ в отсутствие активации.</p>
<p><u>Выбор элемента</u> (Element Selection):</p> $TP(i_l) = a_p + b_p \cdot \log(1 + i_l)$ <p>где a_p и b_p – константы закона Фиттса для оценки времени выбора ЭИ (например, $a_p = 10,3$ и $b_p = 4,8$) для целевого элемента i в точке l.</p>
<p><u>Измерение плотности</u> (Density Measure):</p> $DM = 1 - \frac{1}{a_{frame}} \sum_{i=1}^n a_i$ <p>где a_i и a_{frame} обозначают площадь элемента i и площадь кадра (экрана) соответственно, а n – количество ЭИ в кадре.</p>
<p><u>Соответствие макета</u> (Layout Appropriateness):</p> $LA = 100 \cdot \frac{C_{optimal}}{C_{designed}}$ <p>где $C = \sum_{i \neq j} P_{i,j} \cdot D_{i,j}$, где $P_{i,j}$ и $D_{i,j}$ – частота перехода и расстояние между элементами i и j интерфейса.</p>
<p><u>Баланс</u> (Balance):</p> $BL = 1 - \frac{ BL_{vert} + BL_{hor} }{2}$ <p>где BL_{vert}, BL_{hor} – вертикальный и горизонтальный баланс соответственно.</p> $BL_{hor} = \frac{W_L - W_R}{\max(W_L , W_R)}, \quad BL_{vert} = \frac{W_T - W_B}{\max(W_T , W_B)}$ <p>где L, R, T и B обозначают левый, правый, верхний и нижний края соответственно. W_j – вес j-й стороны ПИ.</p>

Локальный поиск (Local Search):

$TL(i_l) = T_c + \delta \cdot N_{local} + T_{trail}$, где T_c – фиксированное штрафное время (ЭИ обнаружен там, где его не ожидали), δ – фиксированная стоимость тщательного изучения ЭИ, N_{local} – количество проверяемых смежных ЭИ, T_{trail} – фиксированное время, необходимое для отслеживания взгляда пользователя.

Видимость задачи (Task Visibility):

$TV = 100 \cdot \frac{V_i}{S_{total}}$, $\forall i$, где S_{total} – общее количество реализованных шагов для завершения сценария использования, V_i – видимость шага i , от 0 до 1.

Согласование задач (Task Concordance):

$TC = 100 \cdot \frac{2 \cdot D}{N(N-1)}$, где N – количество задач, которые нужно ранжировать, D – индекс несоответствия, то есть количество пар задач, сложность которых распределена правильно, минус количество пар, сложность которых распределена неправильно.

Ввод структурированного текста (Structured Text Entry):

$STE = \frac{1}{n} \sum_i^n Structured_Text_Entry()$, где функция $Structured_Text_Entry()$ возвращает 1, если во входном ЭИ выполняется структурирование текста (маска), и 0 в противном случае; n – общее количество входных ЭИ, принимающих данные в требуемом формате.

Основная эффективность (Essential Efficiency):

$EE = 100 \cdot \frac{S_{essential}}{S_{enacted}}$, где $S_{essential}$ – количество шагов в описании основного (наилучшего) сценария использования, а $S_{enacted}$ – количество шагов, необходимых для выполнения сценария использования с текущим дизайном ПИ.

Встроенная иконка (Built in Icon):

$BI = \frac{1}{n} \sum_i^n Built_in_Icon()$, где $Built_in_Icon()$ возвращает 1, если ЭИ действия отображает иконку, и 0 в противном случае; n – общее количество ЭИ действия.

Значение по умолчанию (Default Value):

$DV = \frac{1}{n} \sum_{i=1}^n a_i$, где a_i – входной элемент со значением по умолчанию, n – общее количество входных ЭИ.

Размер текста (Text Size):

$TxS = \frac{1}{n} \sum_{i=1}^n FontSize_i$, где $FontSize_i$ возвращает 1, если размер шрифта для текстового поля i больше или равен 16 пикселям, в противном случае возвращается 0, а n – количество текстовых полей в интерфейсе.

Горизонтальный (ВН), вертикальный баланс (ВВ):

$Bx = 200 \cdot \frac{W_1}{W_1 + W_2}$, где W_1 – вес первой стороны, W_2 – вес второй стороны. Вес стороны W_i равен количеству использованных пикселей $\times$ расстояние от стороны до центра. Центр – это середина между левым краем самого левого ЭИ и правым краем самого правого ЭИ.

На основании данных метрик **во второй главе** разработана комбинированная модель награды (КМН) для алгоритма пространственной адаптации мобильного ПИ с использованием мультиагентного обучения с подкреплением. Комбинированная модель награды включает комбинацию таких параметров, как целевое состояние среды, метрики удобства использования ПИ и индикатор выполнения задачи.

Согласно теории обучения с подкреплением, агент – это компьютерная программа, которая расположена в некоторой среде и способна автономно действовать и обучаться в этой среде для достижения своих целей.

В модели КМН вознаграждение агента i определяется для каждого параметра k в момент времени $t \neq 0$ по формуле:

$$r_{ik} = 1 - \beta \left| s_k^{(target)} - s_k^{(current)} \right|, \quad (1)$$

где $s_k^{(target)}$ – целевое значение параметра k , характеризующее состояние агента (ЭИ) i и среды, $s_k^{(current)}$ – текущее значение параметра k , β – коэффициент параметра k . Значения s_k' нормализованы в диапазоне от 0 до 1.

Совокупное вознаграждение агента i в модели КМН представляет собой сумму наград по всем k параметрам, u метрикам удобства использования, а также вознаграждение за выполнение задачи:

$$R_i^{UR} = \sum_k r_{ik} + \sum_u r_{iu} + \gamma \mathbb{1}_{successi}, \quad (2)$$

где r_{ik} является наградой агента i для каждого параметра k , r_{iu} является наградой агента i для каждой метрики u из набора юзабилити-метрик (Таблица 1), индикаторная функция $\mathbb{1}_{successi}=1$, если задача агента i успешно завершена. Результат функции $\mathbb{1}_{successi}$ дисконтируется с помощью коэффициента $\gamma \in [0 \dots 1]$.

Среднее значение вознаграждения агента i для каждого параметра k без учета индикаторной функции $\mathbb{1}_{successi}$ также используется для определения качества действий агента: $r_{imean} = \langle r_{ik} \rangle$.

Удобство использования интерфейса автоматически рассчитывается средой, в которой на основе представленных метрик (Таблица 1) генерируется подкрепляющий сигнал вознаграждения, учитываемый в КМН мультиагентной среды после каждого действия, выполненного агентами. Значения метрик удобства использования r_{iu} находятся в диапазоне от 0 до 1, где $r_{iu} \rightarrow 0$ указывает на низкое качество интерфейса, $r_{iu} \rightarrow 1$ – на высокое качество.

Математическая модель АПИ для алгоритма пространственной адаптации ПИ на основе теоретико-множественного подхода включает: множества контекстов использования, типов задач и интересов пользователя, событий, правил адаптации, ЭИ, метрик удобства использования, отношений и функций, определяющих поведение системы. Модель АПИ для алгоритма пространственной адаптации представляется кортежем: $AUI_{spatial} = (C, T, I, E, R, L, M, \Phi)$, где:

1. $C = \{c_1, c_2, \dots, c_m\}$ – множество контекстов использования, определяемых кортежем $c_i = (t_{day}, p_{geo}, d_{ilm}, o_{dev}, s_{size}, res_{scr})$, где t_{day} – время

суток, p_{geo} – геопозиция, d_{ilm} – освещенность, o_{dev} – ориентация устройства (портрет/альбом), s_{size} – размер экрана, res_{scr} – разрешение экрана.

2. $T = \{t_1, t_2, \dots, t_y\}$ – множество типов задач пользователя (чтение, просмотр, ввод, поиск, навигация и другие).

3. $I = \{i_1, i_2, \dots, i_x\}$ – множество интересов пользователя, заданное на основе истории взаимодействий (темный стиль ПИ, жест «свайп», мультимедийное содержимое).

4. $E = \{e_1, e_2, \dots, e_k\}$ – множество событий, инициирующих адаптацию.

5. $R = \{r_1, r_2, \dots, r_t\}$ – множество правил адаптации, определяемых кортежем $r_i = (s_{obj}, s_{trg}, s_{dyn}, s_{src})$, где s_{obj} – объект адаптации (компоновка, размеры, визуальное оформление, интерактивность), s_{trg} – цель адаптации (под задачу, персонализация, доступность), s_{dyn} – тип динамики адаптации (реактивная, проактивная), s_{src} – источник данных (датчики устройства, поведение пользователя, ОС).

6. $L = \{l_1, l_2, \dots, l_p\}$ – множество ЭИ, задающих макет ПИ. Каждый элемент $l_i \in L$ представляет собой ЭИ и его свойства: расположение, размер и стиль. Существует L_{cur} – текущий макет, L_{target} – прогнозируемый целевой макет ПИ, L_{new} – макет после адаптации ПИ, L_{def} – макет по умолчанию (до первой адаптации).

7. $M = \{m_1, m_2, \dots, m_r\}$ – множество метрик удобства (Таблица 1). Метрики описывают качество взаимодействия – удобство использования (время на задачу, количество ошибок, карта кликов). Например, метрики вычисляются для текущего контекста и макета ПИ: $M_{cur} = \phi_{URM}(C, L_{cur})$.

8. Φ – множество отношений, определяющих поведение системы, такие как: $\phi_R: C \times T \times I \times E \rightarrow R$ – отношение контекстного анализа; $\phi_{URM}: (C, L) \rightarrow M$ – вычисления юзабилити-метрик для макета ПИ; $\phi_{pr}: (L, C, T) \rightarrow L_{pr}$ – определение приоритетных ЭИ; $\phi_L: (L, L_{pr}, M, R) \rightarrow L_{target}$ – прогноз целевого макета ПИ; $\phi_{val}: L \rightarrow \{True, False\}$ – проверка корректности макета; $\phi_{tr}: L_{cur} \times L_{target} \rightarrow \{True, False\}$ – проверка разрешимости перехода; $\phi_{Map}: (L_{cur}, L_{target}) \rightarrow L_{new}$ – применение изменений (ПИ); $\phi_U: (T, I, L_{new}, L_{cur}) \rightarrow \{True, False\}$ – проверка корректности L_{new} в сравнении с L_{cur} .

В Алгоритме 1 представлена последовательность выполнения цикла пространственной адаптации мобильного ПИ. В момент запуска системы, перед входом в алгоритм, выполняется инициализация множеств C и L_{cur} , то есть система находится в состоянии (C, L_{cur}) . Далее, в течение некоторого времени, формируется множество задач T и интересов I , характеризующих поведение пользователя. При наступлении нового события $e \in E$ выполняется Алгоритм 1, и повторяется при срабатывании одного из событий E , например: изменение контекста; переключение приложения; истечение заданного временного интервала (системное событие).

Алгоритм 1. Пространственная адаптация мобильного ПИ на основе мультиагентного обучения с подкреплением и комбинированной модели награды

1. $R_{cur} = \phi_R(C, T, I, E)$ // Правила адаптации
2. $M_{cur} = \phi_{URM}(C, L_{cur})$ // Юзабилити-метрики из КМН
3. $L_{pr} = \phi_{pr}(L_{cur}, C, T)$ // Приоритетные ЭИ
4. $L_{target} = \phi_L(L_{cur}, L_{pr}, M_{cur}, R_{cur})$ // Прогноз целевого макета ПИ
5. **Если** $L_{target} \neq L_{cur}$ и $\phi_{val}(L_{target}) = True$, **то**: // Валидация макета ПИ
 Если $\phi_{tr}(L_{cur}, L_{target}) = True$, **то**: // Разрешимость перехода
 $L_{new} = \phi_{Map}(L_{cur}, L_{target})$ // Адаптация ПИ
 Формирование $i_{cur} \in I$ // Интересы и отклик пользователя
6. **Если** $L_{new} = L_{cur}$ или $L_{target} = L_{cur}$ или $\phi_{val}(L_{target}) = False$, **то**:
 // Адаптация не производится, система продолжает мониторинг
 Возврат
7. **Если** $\phi_U(T, I, L_{new}, L_{cur}) = True$, **то**:
 Вычисление награды, обучение агентов, обновление функции ϕ_L
 Иначе:
 $L_{new} = \phi_{Map}(L_{cur}, L_{def})$ // Отмена адаптации: $L_{new} = L_{def}$

Верхняя оценка вычислительной сложности Алгоритма 1 составляет $O(n \cdot \log n)$, где n — количество ЭИ в макете.

Юзабилити-тестирование по методике QUIS 7.0 (Questionnaire for User Interface Satisfaction) мобильного ПИ, полученного в результате адаптации на основе мультиагентных алгоритмов IQL и MADDPG, продемонстрировало, что алгоритм IQL на 6% эффективнее алгоритма MADDPG в задаче пространственной адаптации мобильного ПИ.

Проведено экспериментальное исследование Алгоритма 1 с моделью КМН по критерию эмпирической оценки времени, затрачиваемого на поиск и выбор интересующих пользователя ЭИ, в сравнении с двумя другими ПИ: статичный и на основе частотной адаптации. Статичный ПИ представляет собой сетку, в которой все ЭИ неподвижны и расположены случайным образом. В интерфейсе на основе частотной адаптации ЭИ перемещаются в зависимости от выбранного приоритетного положения на экране и частоты использования ЭИ. Дисперсионный анализ времени поиска и выбора ЭИ представлен на Рисунке 1. Сравнение трех типов мобильных ПИ – адаптивного, статичного и частотного – выявило, что адаптивный интерфейс (на основе КМН) на 9% эффективнее частотного АПИ по времени поиска и выбора ЭИ.

В третьей главе представлен алгоритм образной адаптации мобильного ПИ, который позволяет персонализировать визуальные ЭИ (кнопки, поля, списки), подстраивая их форму, цвет и стиль под предпочтения пользователя – например, для улучшения доступности, настроения или контекста использования. Выполнен анализ генеративных нейросетевых моделей: вариационный автоэнкодер (Variational Autoencoder, VAE), генеративно-

состязательная сеть (Generative Adversarial Network, GAN) и стабильная диффузия (Stable Diffusion, SD). Для сравнения VAE, GAN и SD приводится анализ полученных изображений ПИ на основе метрик удобства использования (Таблица 1).

Математическая модель АПИ для алгоритма образной адаптации представляется кортежем: $AUI_{form} = (C, T, I, L, M, E, H, Pr, G, \Phi)$, где:

1. Множества C, T, I, L, M, E задаются аналогично рассмотренному в главе 2 алгоритму пространственной адаптации $AUI_{spatial}$.

2. H – множество истории взаимодействий пользователя, в котором $h = (taps, time, profile) \in H$, где $taps$ – клики по ЭИ, $time$ – время просмотра содержимого, $profile$ – предпочтения из профиля.

3. Pr – множество текстовых описаний ЭИ, где $pr \in Pr$ – текстовое описание (prompt) ЭИ или функции.

4. G – множество генерируемых образов ЭИ, каждый из которых представляет собой кортеж $g = (style, color, shape, size, theme, os) \in G$, где $style$ – стиль (плоский/минимализм), $color$ – цветовая схема (HSV/RGB/hex), $shape$ – форма, $size$ – масштаб, $theme$ – светлая/темная тема, os – целевая ОС.

5. Φ – отношения, определяющие поведение системы, такие как: $\phi_T: C \rightarrow T$ – классификация задачи пользователя на основе контекста использования C ; $\phi_I: H \rightarrow I$ – определение интересов пользователя из истории взаимодействий; $\phi_L: C \times T \times I \rightarrow L_{target}$ – выбор целевых ЭИ для адаптации; $\phi_{Pr}: L \times C \times T \times I \rightarrow Pr$ – создание текстового описания образа ЭИ с требуемыми параметрами: l – ЭИ и его характеристики (цвет, форма, размер); c – контекст (стиль, тема, целевая ОС); t – задачи пользователя, i – интересы пользователя: $pr_l = \phi_{Pr}(l, c, t, i)$; $\phi_G: L \times Pr \rightarrow G$ – формирование нового образа генеративной моделью VAE/GAN/SD на основе ЭИ и текстового запроса. Таким образом, ϕ_G является ядром алгоритма и реализуется с использованием моделей; $\phi_M: G \rightarrow M$ – ранжирование образов по юзабилити-метрикам; $\phi_{opt}: G \rightarrow G_{opt}$ – оптимизация графики образа ЭИ; $\phi_{Map}: L \times G_{opt} \rightarrow L_{updated}$ – преобразование макета ПИ для замены образов ЭИ, где $l \in L$ – образ до замены, и $g \in G_{opt}$ – образ после замены; $\phi_O: L \rightarrow \mathbb{R}$ – оценка производительности: оценка времени рендеринга (отрисовки графики) t_r , которое должно удовлетворять условию $t_r < 50$ мс.

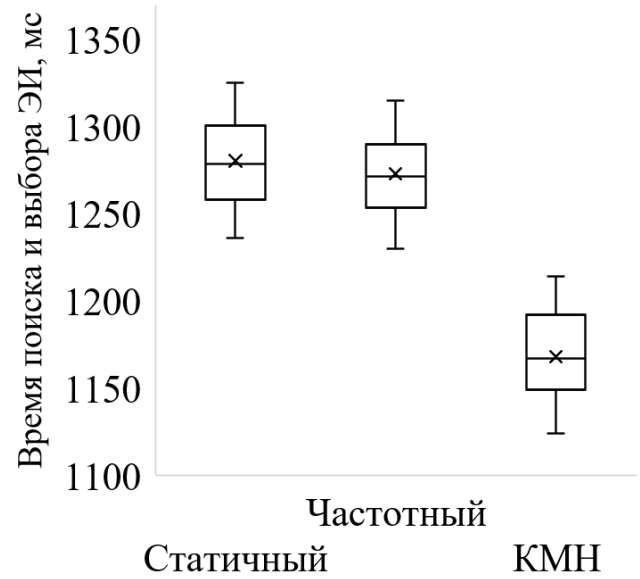


Рисунок 1 – Сравнение трех типов мобильных ПИ

В Алгоритме 2 представлена последовательность выполнения цикла образной адаптации мобильного ПИ. В начальный момент, перед входом в алгоритм, выполняется инициализация множеств C и H .

Алгоритм 2. Образная адаптация мобильного ПИ на основе генеративных моделей VAE/GAN/SD

1. $c_{cur} = (t_{day}, p_{geo}, d_{ilm}, o_{dev}, s_{size}, res_{scr})$, $c_{cur} \in C$ // Текущий контекст
2. $t_{cur} = \phi_T(c_{cur})$, $t_{cur} \in T$ // Задача пользователя
3. $i_{cur} = \phi_I(H)$, $i_{cur} \in I$ // Интересы пользователя
4. $L_{target} = \phi_L(c_{cur}, t_{cur}, i_{cur})$ // Множество ЭИ и макет ПИ
5. Для каждого $l \in L_{target}$:
 - 5.1. $pr_l = \phi_{Pr}(l, c_{cur}, t_{cur}, i_{cur})$, $pr_l \in Pr$ // Текстовое описание ЭИ
 - 5.2. $g_{cand} = \phi_G(l, pr_l)$, $g_{cand} \in G_{cand}$ // Образ ЭИ из VAE/GAN/SD
 - 5.3. $M = \phi_M(G_{cand})$ // Юзабилити-метрики образов ЭИ
 - 5.4. $g_{best} = \operatorname{argmax}_{g \in G_{cand}}(M)$ // Лучший образ ЭИ
 - 5.5. Если $\phi_M(g_{best}) < 0.8$, то:
 - $g_{final} = g_{std} \in G_{std}$ // Стандартный образ
 - Иначе:
 - $g_{final} = g_{best} \in G_{cand}$ // Сгенерированный образ
 - 5.6. $g_{opt} = \phi_{opt}(g_{final})$, $g_{opt} \in G_{opt}$ // Оптимизация образов для МУ
6. $L_{new} = \phi_{Map}(L_{target}, G_{opt})$ // Новое множество ЭИ и макет ПИ
7. Для каждого $l \in L_{new}$:
 - 7.1. $t_r = \phi_O(l)$ // Время рендеринга графики
 - 7.2. Если $t_r \geq 50$ мс, то:
 - $l \rightarrow l_{bad} \in L_{bad}$ // Непрошедшие проверку ЭИ
 - 7.3. Для L_{bad} повторить шаги 5.2–5.6 с ограничениями графики
8. Добавить в H информацию о взаимодействии пользователя с L_{new}

Верхняя оценка вычислительной сложности Алгоритма 2 составляет $O(n \cdot H \cdot W)$, где n – количество ЭИ в макете, $H \cdot W$ – разрешение выходного изображения. Юзабилити-тестирование по методике QUIS 7.0 мобильного ПИ, полученного с использованием алгоритма образной адаптации продемонстрировало, что данный АПИ на 7% эффективнее статичного ПИ.

В четвертой главе представлен алгоритм функциональной адаптации мобильного ПИ, в основе которого лежит использование программных средств обнаружения и извлечения сервисов ОС, анализ запущенных мобильных приложений для распознавания и определения доступных функций. Реализована оригинальная сверточная нейросетевая модель для определения категории мобильного приложения на основе изображения ПИ.

Математическая модель АПИ для алгоритма функциональной адаптации представляется кортежем: $AUI_{func} = (C, T, I, E, A, S, F, L, D, P, N, \Phi)$, где:

1. Множества C, T, I, E аналогично рассмотренному в главе 2 алгоритму пространственной адаптации $AUI_{spatial}$.

2. $A = \{a_1, a_2, \dots, a_i\}$ – множество приложений, где a_i – приложение, запущенное в ОС. Каждое приложение описывается кортежем $a_i = (proc_i, meta_i)$, где $proc_i$ – процесс приложения, $meta_i$ – метаданные приложения.

3. $S = \{s_1, s_2, \dots, s_r\}$ – множество сервисов ОС, где s_r – сервис ОС, доступный для использования в приложениях.

4. $F = \{f_1, f_2, \dots, f_j\}$ – множество функций приложений, где f_j – функция приложения. Функция определяется кортежем $f_j = (a_i, intent_j, action_j, text_j, conf_j)$, где a_i – источник функции (приложение, процесс ОС), $intent_j$ – объект для передачи намерения выполнить действие, $action_j$ – действие, $text_j$ – текстовое описание функции, $conf_j$ – оценка достоверности распознавания функции. Подмножество достоверных функций: $F_{valid} = \{f_j \in F \mid conf_j > \theta_f\}$, где $\theta_f = 0.8$.

5. $L = \{l_1, l_2, \dots, l_k\}$ – множество ЭИ и макет ПИ. Каждый ЭИ задается кортежем: $l_k = (pos_k, type_k, text_k, conf_k)$, где pos_k – координаты, $type_k$ – тип ЭИ, $text_k$ – текст ЭИ, $conf_k$ – вероятность обнаружения ЭИ. Подмножество достоверных ЭИ: $L_{valid} = \{l_k \in L \mid conf_k > \theta_l\}$, где $\theta_l = 0.8$.

6. $D = \{d_1, d_2, \dots, d_z\}$ – множество эмбедингов (векторное представление объекта), где d_z – вектор ЭИ или функции. Формирование эмбедингов функций: $d_{f_j} = \phi_e(f_j)$, где $f_j \in F_{valid}$ – функция, $d_{f_j} \in \mathbb{R}^d$ – вектор функции. Формирование эмбедингов ЭИ: $d_{l_k} = \phi_e(l_k)$, где $l_k \in L_{valid}$ – ЭИ, $d_{l_k} \in \mathbb{R}^d$ – вектор ЭИ.

7. $P \subseteq F \times L$ – подмножество сопоставленных функций и ЭИ, где $(f_j, l_k) \in P$, если функция f_j соответствует элементу интерфейса l_k . Для определения P используется отношение семантического сходства: $sim(f_j, l_k) = \frac{\phi_e(f_j) \cdot \phi_e(l_k)}{\|\phi_e(f_j)\| \|\phi_e(l_k)\|}$, где $f_j \in F$ и $l_k \in L$. Подмножество сопоставленных ЭИ: $P = \{(f_j, l_k) \mid sim(f_j, l_k) > \theta_p\}$, где $\theta_p = 0.8$.

8. $N = \{n_1, n_2, \dots, n_q\}$ – множество новых адаптированных ЭИ, где n_q – ЭИ, добавляемый в новый (целевой) ПИ. Каждый новый ЭИ определяется: $n_q = (pos_q, f_j, l_k)$, где pos_q – позиция, l_k – ЭИ и соответствующая функция f_j . Формирование новых ЭИ: $N = \{n_q = (pos_q, f_j, l_k) \mid (f_j, l_k) \in P\}$. Если функция, соответствующая ЭИ в N становится неактивной, то формируется множество неактивных ЭИ: $N_{inactive} = \{n_q \in N \mid t_{inactive}(n_q) > 30 \text{ сек.}\}$, где $t_{inactive}$ – время отсутствия использования функции.

9. Φ – множество отношений, определяющих поведение системы, такие как: $\phi_S: A \times S \rightarrow S_{act}$ – обнаружение активных сервисов ОС; $\phi_L: A \rightarrow L$ – извлечение макета ПИ запущенных приложений; $\phi_F: (A, L, S_{act}) \rightarrow F$ – обнаружение функций; $\phi_e: (F \cup L) \rightarrow D$ – формирование эмбедингов; $D \subseteq \mathbb{R}^d$, где d – размерность эмбединг-пространства; $\phi_P: (F_{valid}, L_{valid}) \rightarrow P$ – определение соответствия функций и ЭИ; $\phi_N: P \rightarrow N$ – формирование новых

ЭИ; $\phi_{Map}: (L, N) \rightarrow L_{new}$ – применение изменений ПИ; $\phi_O: (CPU, latency) \rightarrow \Delta t$ – проверка производительности системы и вычисление интервала оптимизации.

В Алгоритме 3 представлена последовательность выполнения итерации цикла функциональной адаптации мобильного ПИ, операций над множествами и функциями модели. Периодичность выполнения алгоритма задается интервалом $\Delta t = 5$ с, а также при срабатывании одного из событий E , например, при изменении контекста использования, взаимодействии пользователя с ПИ или переключении приложения.

Алгоритм 3. Функциональная адаптация мобильного ПИ на основе сверточной нейросетевой модели	
1. Инициализация множеств A, S	
2. $S_{act} = \phi_S(A, S)$	// Активные сервисы ОС
3. $L_{cur} = \phi_L(A)$	// Извлечение ЭИ и макета ПИ
4. $F = \phi_F(A, L_{cur}, S_{act})$	// Распознавание функций
5. $P = \phi_P(F_{valid}, L_{valid})$	// Сопоставление функций с ЭИ
6. $N = \phi_N(P)$	// Формирование новых ЭИ
7. $N_{inactive} = \{n_q \in N \mid t_{inactive}(n_q) > 30 \text{ сек.}\}$	// Неактивные ЭИ
8. $N = N \setminus N_{inactive}$	// Удаление неактивных ЭИ
9. $L_{new} = \phi_{Map}(L_{cur}, N)$	// Добавление новых ЭИ
10. $\Delta t = \phi_O(CPU, latency)$	// Проверка производительности

Верхняя оценка вычислительной сложности Алгоритма 3 составляет $O(n^2 \cdot d)$, где n – количество ЭИ в макете, d – размерность пространства эмбедингов. Юзабилити-тестирование по методике QUIS 7.0 мобильного ПИ, полученного с использованием алгоритма функциональной адаптации продемонстрировало, что данный АПИ на 9% эффективнее статичного ПИ.

В пятой главе рассматривается программная архитектура АПИ МУ на основе сервиса межпроцессного взаимодействия D-Bus (Desktop Bus) – механизм, позволяющий приложениям и системным службам обмениваться данными и координировать работу.

АПИ МУ, основанный на сервисе D-Bus, включает реализацию алгоритма пространственной, образной и функциональной адаптации, с использованием алгоритма IQL мультиагентного обучения с подкреплением, комбинированной модели награды, генеративной и сверточной нейросетевой модели (Рисунок 2).

Обработка запросов через сервис D-Bus включает объекты $/ru/template/AppX$, где $X=1..N$ – индекс объекта для соответствующего приложения, что позволяет клиенту предоставлять интерфейс/методы через D-Bus – конечная точка связи, созданная клиентом и существующая в контексте его подключения. Путь объекта определяется идентификатором клиента (приложения). При этом, клиент может создавать один или множество объектов.

Реализован прототип приложения Аврора с использованием менеджера D-Bus, который был разработан в соответствии с необходимыми характеристиками ОС. На Рисунке 3 приводятся результаты мультиагентного обучения с

подкреплением (IQL) при выполнении пространственной адаптации и вычисление вознаграждения агентов с использованием комбинированной модели награды в ОС Аврора.



Рисунок 2 – Программная архитектура АПИ на основе сервиса межпроцессного взаимодействия D-Bus для ОС Аврора

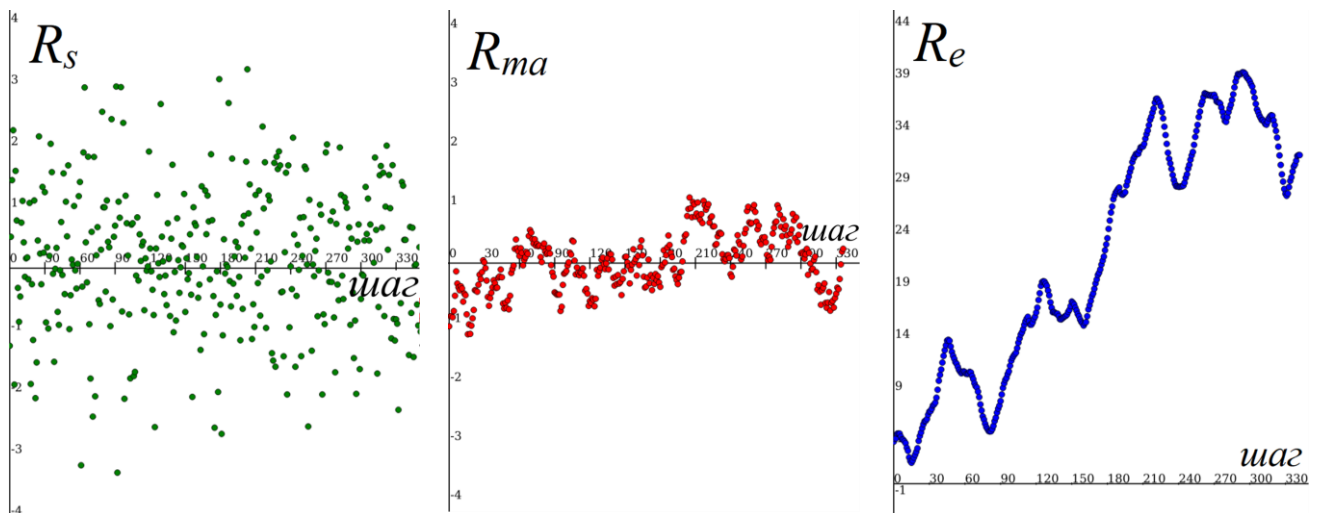


Рисунок 3 – Суммарная награда трех агентов в течение эпизода 350 шагов. Во всех графиках вертикальная ось – Награда (Reward), горизонтальная – номер шага (Step): R_s – суммарная награда на каждом шаге, R_{ma} – скользящее среднее суммарной награды, R_e – совокупная награда в результате всех шагов эпизода

В заключении приведены основные результаты диссертационного исследования:

1. Выполнен обзор и анализ существующих методов адаптации мобильного ПИ. Рассмотрены методы оценки удобства использования ПИ на основе подходов квантификации и юзабилити-метрик.

2. Разработан алгоритм пространственной адаптации мобильного ПИ, который корректирует расположение и размеры ЭИ в соответствии с контекстом использования, выполняемой задачей и интересами пользователя. Проведен анализ качества и удобства использования АПИ, полученного в результате адаптации на основе алгоритмов IQL и MADDPG, а также юзабилити-тестирование по методике QUIS 7.0.

3. Разработан алгоритм образной адаптации на основе нейросетевых моделей VAE/GAN/SD для генерирования визуального представления элементов мобильного ПИ. Выполнен анализ полученных изображений ПИ на основе метрик удобства использования, а также юзабилити-тестирование для оценки пользовательского опыта при использовании АПИ, реализующего алгоритм образной адаптации в сравнении со статичным ПИ (QUIS 7.0).

4. Разработан алгоритм функциональной адаптации на основе сверточной нейросетевой модели для определения категории мобильного приложения. Выполнено юзабилити-тестирование (QUIS 7.0) пользовательского опыта для АПИ алгоритма функциональной адаптации.

5. Представлена программная архитектура АПИ для отечественной мобильной ОС Аврора на основе сервиса межпроцессного взаимодействия D-Bus, включающая программные средства алгоритмов функциональной, образной и пространственной адаптации мобильного ПИ. Проведены экспериментальные исследования и внедрение предложенных программных средств АПИ МУ, позволяющие оценить результативность предлагаемых алгоритмических и программных решений.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Видьманов Д. А. Программные средства адаптивного пользовательского интерфейса на основе межпроцессного взаимодействия в отечественной мобильной ОС Аврора // Информатизация и связь. 2025. № 4. С. 84–96. (1,5 п.л.)

2. Видьманов Д. А., Алфимцев А. Н. Адаптивный мобильный пользовательский интерфейс // Промышленные АСУ и контроллеры. 2022. № 7. С. 11–19. (1,0 п.л./0,5 п.л.)

3. Видьманов Д. А. Визуализация академической успеваемости в электронных таблицах облачных сервисов с использованием веб-приложения и мобильных устройств // Приборы и системы. Управление, контроль, диагностика. 2022. № 3. С. 20–30. (1,3 п.л.)

4. Markevich D., Vidmanov D. Reinforcement Learning for a Playground on the Aurora Mobile OS: An Example Using the MuJoCo Ant Framework // 2026 International Conference on Integrated Intelligence and Cognitive Engineering (ICIICE), Dubai, United Arab Emirates, 2026, P. 1–6. (1,1 п.л./0,55 п.л.)

5. Kostinich K. V., Vidmanov D. A. Generative user interface for the mobile apps: image synthesis with VAEs, GANs, and Stable Diffusion // 2025 7th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE). IEEE, 2025. P. 1–6. (1,1 п.л./0,55 п.л.)
6. Markevich D. S., Vidmanov D. A. Implementation of a neural network model on the Aurora operating system using ARM architecture // 2025 7th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE). IEEE, 2025. P. 1–5. (1,0 п.л./0,5 п.л.)
7. Vidmanov D., Alfimtsev A. Mobile user interface adaptation based on usability reward model and multi-agent reinforcement learning // Multimodal Technologies and Interaction. 2024. Vol. 8. №. 26. P. 1–21. (2,7 п.л./1,35 п.л.)
8. Vidmanov D. A., Alfimtsev A. N. Intelligent analysis of the user interface of mobile applications to search for functions and services // 2024 6th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE). IEEE, 2024. P. 1–5. (1,0 п.л./0,5 п.л.)
9. Vidmanov D. A., Alfimtsev A. N. MARLMUI: Multi-agent reinforcement learning approach in Mobile Adaptive User Interface // 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE). IEEE, 2023. Vol. 5. P. 1–5. (1,0 п.л./0,5 п.л.)
10. Любимов Д. Н., Ким Д. К., Видьманов Д. А. Разработка пользовательского интерфейса с помощью специализированных библиотек на примере планировщика задач // Внедрение современных конструкций и передовых технологий в путевое хозяйство. 2024. Т. 20. С. 74–80. (0,8 п.л./0,27 п.л.)
11. Костинич К. В., Видьманов Д. А. Мобильный пользовательский интерфейс на основе генеративных моделей // Робототехника и искусственный интеллект. 2023. С. 136–141. (0,7 п.л./0,35 п.л.)
12. Vidmanov D. A., Alfimtsev A. N. Application of Dec-POMDP model to the multi-agent user interface // Интеллектуальные системы 4-й промышленной революции. 2023. С. 122–123. (0,2 п.л./0,1 п.л.)
13. Видьманов Д. А., Алфимцев А. Н. Адаптация пользовательского интерфейса на основе мультиагентного обучения с подкреплением // Фундаментальные, поисковые, прикладные исследования и инновационные проекты. 2022. С. 524–528. (0,6 п.л./0,3 п.л.)
14. Программа адаптивного мобильного пользовательского интерфейса на основе обучения с подкреплением: свидетельство о государственной регистрации программы для ЭВМ № 2023683047 РФ. / Д. А. Видьманов, А. Н. Алфимцев; заявл. 24.10.2023; опубл. 02.11.2023. Бюлл. № 2023682038.
15. Технология предиктивной адаптации мобильного пользовательского интерфейса: свидетельство о государственной регистрации программы для ЭВМ № 2022663865 РФ. / Д. А. Видьманов; заявл. 08.07.2022; опубл. 20.07.2022. Бюлл. № 2022663377.