

МЕТОДЫ АДАПТИВНОГО ФОРМИРОВАНИЯ ГРАФИЧЕСКИХ ИНТЕРФЕЙСОВ В ПРОГРАММНЫХ ПЛАТФОРМАХ ЭЛЕКТРОННОЙ КОММЕРЦИИ

METHODS OF ADAPTIVE GRAPHICAL INTERFACE DEVELOPMENT IN E-COMMERCE SOFTWARE PLATFORMS

**M. Serov
O. Romashkova
T. Dobrynina**

Summary. The article presents models and methods for designing e-commerce software systems based on interactive graphical user interfaces. The relevance of the research is driven by the increasing complexity of user scenarios and growing requirements for interface adaptability across heterogeneous access devices. An architectural model of an interface-oriented e-commerce system is proposed, and a methodology for experimental evaluation of resource and functional characteristics. A prototype including a state management model, a resource-efficiency controller, and a component library has been developed. Experimental results demonstrate reduced interface latency and improved rendering performance. The findings can be applied to the design of commercial web and mobile applications.

Keywords: graphical user interface, e-commerce, rendering optimization, UX models, adaptive UI, client-side performance.

Серов Максим Валерьевич

Аспирант, ГАОУ ВО Московский городской педагогический университет
08maks80@bk.ru

Ромашкова Оксана Николаевна

Доктор технических наук, профессор,
ФГБОУ ВО Российская академия народного хозяйства
и государственной службы при Президенте РФ, г. Москва
ox-rom@yandex.ru

Добрынина Татьяна Николаевна

Кандидат технических наук, доцент, ГАОУ ВО
Московский городской педагогический университет
ermaktat@bk.ru

Аннотация. В статье представлены модели и методы проектирования программных систем электронной коммерции, ориентированных на использование интерактивных графических интерфейсов. Актуальность работы определяется усложнением пользовательских сценариев и ростом требований к адаптивности интерфейсов в условиях высокой вариативности устройств доступа. Предложены архитектурная модель интерфейс-ориентированной e-commerce системы, а также методика экспериментальной оценки ресурсных и функциональных характеристик. Разработан прототип, включающий модель состояния интерфейса, контроллер ресурсной эффективности и компонентную библиотеку. Экспериментальная проверка продемонстрировала снижение латентности интерфейса и повышение производительности рендеринга. Результаты могут быть применены при проектировании коммерческих веб- и мобильных приложений.

Ключевые слова: графический интерфейс, электронная коммерция, рендеринг интерфейса, UX-модели, адаптивные интерфейсы, оптимизация клиентских приложений.

Введение

Современные программные платформы электронной коммерции характеризуются высокой динамичностью данных, сложностью пользовательских сценариев и необходимостью адаптации интерфейса под широкий спектр устройств. Растущие требования к удобству взаимодействия, времени отклика и стабильности работы приводят к усложнению архитектуры клиентских приложений. Визуальные компоненты становятся всё более насыщенными, увеличивается глубина вложенности элементов, возрастает количество контекстных обновлений и интерактивных блоков. В таких условиях обеспечение эффективного формирования и адаптации графического интерфейса становится одной из ключевых задач разработки e-commerce систем.

Современные исследования в области архитектуры программных систем подчёркивают, что без формализованной модели состояния интерфейса и оптимизированных методов обновления визуальных компонентов невозможно обеспечить предсказуемое и устойчивое поведение интерфейса. Особую значимость приобретают компонентные архитектуры, позволяющие отделить визуальное представление от логики и обеспечить масштабируемость структуры интерфейса. Российские и зарубежные работы также указывают на необходимость применения методов частичного рендеринга, оптимизации количества перерисовок и адаптивных моделей отображения компонентов на устройствах с различными характеристиками экрана и вычислительными ресурсами [1].

При этом в настоящее время сохраняется ряд проблем. Во-первых, отсутствуют достаточно обобщённые формальные модели, позволяющие описывать поведение интерфейсных состояний в условиях динамических изменений данных, вероятностных переходов и множественных контекстов использования. Во-вторых, недостаточно разработаны методы адаптации интерфейса, учитывающие одновременно пользовательский контекст, характеристики устройства и бизнес-логику приложения. В-третьих, существующие экспериментальные методики оценки производительности UI часто не учитывают особенности e-commerce сценариев, включающих фильтрацию каталога, сортировку, навигацию, работу корзины и оформление заказа.

Таким образом, возникает необходимость разработки комплексного подхода, включающего архитектурные, математические модели, а также методики экспериментальной оценки эффективности интерфейсных решений.

Цель исследования — разработать архитектурную модель и методы проектирования интерфейс-ориентированных программных систем электронной коммерции, а также выполнить экспериментальную оценку эффективности предложенных решений.

Для достижения поставленной цели решаются следующие задачи:

1. Формализовать модель пользовательских сценариев и состояния графического интерфейса e-commerce систем;
2. Разработать методы генерации, адаптации и оптимизации графических интерфейсов;
3. Построить экспериментальный стенд и провести оценку эффективности предложенных моделей.

Предложенный подход позволяет систематизировать процессы проектирования интерфейс-ориентированных e-commerce приложений и обеспечивает основу для разработки высокопроизводительных, адаптивных и устойчивых клиентских решений.

Модели проектирования интерфейс-ориентированных систем электронной коммерции

Проектирование интерфейс-ориентированных программных систем электронной коммерции основывается на формализации пользовательских сценариев и архитектурных принципов, обеспечивающих согласованность работы всех интерфейсных компонентов. Ключевым элементом такой формализации является модель состояний интерфейса, представляющая структуру e-commerce приложения в виде множества возможных экранов и состояний взаимодействия

$$S = \{S_0, S_1, \dots, S_n\}, \quad (1)$$

где S_0 соответствует начальному состоянию (главный экран или каталог), а остальные состояния описывают карточку товара, корзину, оформление заказа, профиль пользователя и другие элементы.

Поведение пользователя в системе описывается через множество переходов

$$T = \{t_1, t_2, \dots, t_m\}, \quad (2)$$

где каждый переход $t_i: S_j \rightarrow S_k$ соответствует действию пользователя (выбор товара, применение фильтров, добавление в корзину, смена параметров поиска и др.). Эти переходы формируют детерминированный или вероятностный граф интерфейсного поведения. В ряде случаев удобно использовать марковскую модель вероятностей переходов между состояниями

$$P(s_j, s_i) = \frac{N(s_i \rightarrow s_j)}{N(s_j)}, \quad (3)$$

что позволяет количественно оценивать наиболее востребованные пути взаимодействия и узкие места интерфейса.

Пользовательская сессия описывается последовательностью пар «состояние–действие»

$$U = \{(s_0, a_0), (s_1, a_1), \dots, (s_k, a_k)\}, \quad (4)$$

что делает возможной формализацию типовых сценариев и построение моделей поведения пользователей для последующей оптимизации интерфейса. Для сложных интерфейсов вводится иерархия состояний, включающая базовые и вложенные уровни

$$S = S^{(base)} \cup S^{(nested)}, \quad (5)$$

где вложенные состояния описывают модальные окна, всплывающие панели, слои навигации и блоки рекомендаций.

Архитектура интерфейс-ориентированных e-commerce систем строится на компонентном подходе. В рамках используемой модели MVI (Model–View–Intent) интерфейс разбивается на независимые компоненты, взаимодействующие через единое дерево состояний. Такой подход обеспечивает предсказуемость поведения интерфейса, удобство тестирования и возможность генерации интерфейсных конфигураций на основе изменяющихся данных.

Контекстная адаптация интерфейса описывается функцией

$$UI = F(C, D, U), \quad (6)$$

где C — контекст пользователя (история действий, предпочтения), D — параметры устройства (размер экрана, DPI, ориентация, мощность процессора), U — бизнес-правила.

Дополнительно вводится метрика плотности интерфейса, позволяющая оценить нагрузку на визуальный слой

$$D = \frac{n_c}{A}, \tag{7}$$

где n_c — число визуальных компонент, а A — площадь активной области экрана. Возрастание плотности обычно приводит к увеличению задержек интерфейса и количеству перерисовок.

Функция адаптации состояния интерфейса при взаимодействии пользователя формально записывается следующим образом

$$s_{t+1} = G(s_t, e), \tag{8}$$

где e — событие (клик, свайп, фильтр, ввод данных).

Эта модель определяет, какие компоненты должны быть обновлены, и обеспечивает минимизацию операций рендеринга при динамических изменениях данных.

Оптимизация интерфейса формализуется как задача минимизации совокупной стоимости рендеринга

$$UI^* = \arg \min_{UI} (\alpha \cdot latency + \beta \cdot complexity + \gamma \cdot error), \tag{9}$$

где параметры α, β, γ определяют относительную значимость скорости отклика, сложности визуальной структуры и количества ошибок интерфейсных элементов.

Таким образом, объединённая модель проектирования интерфейс-ориентированных e-commerce систем включает: графовую модель состояний, иерархию компонент, архитектурную модель MVI, механизм адаптации интерфейса и математическую формализацию критериев оптимизации. Эти элементы создают фундамент для разработки эффективных методов генерации и обновления пользовательских интерфейсов, а также для экспериментальной оценки их производительности в условиях реальной нагрузки.

Методы формирования и адаптации графических интерфейсов

Методы проектирования графических интерфейсов в системах электронной коммерции направлены на обеспечение высокой производительности, адаптивности и предсказуемости поведения интерфейса при динамически изменяемых данных. Базой для разработки мето-

дов служит компонентная модель интерфейса, в которой каждый визуальный элемент описывается в виде независимого блока, обладающего собственным состоянием и набором входных параметров [2]. Формирование интерфейсного состояния опирается на входной набор данных

$$P = \{p_1, p_2, \dots, p_k\}, \tag{10}$$

где p_1 — определяет параметры товара, p_2 — фильтры и сортировки, p_3 — содержимое корзины, а остальные параметры относятся к структуре пользовательского пути и характеристикам устройства.

Процесс генерации интерфейса включает последовательность шагов: сбор параметров, выбор шаблона интерфейса, формирование набора компонент, генерацию дерева визуальных элементов, проверку ограничений вычислительных ресурсов и финальный рендеринг [3]. Этот процесс можно формализовать как отображение

$$UI = G(P), \tag{11}$$

где G — функция композиции компонент, определяющая структуру интерфейса на основе входных данных. Учитывая высокую изменчивость пользовательских действий и данных, важнейшим требованием является способность интерфейса обновляться без полной перерисовки экранов.

Адаптация интерфейса к действиям пользователя реализуется через функцию перехода между интерфейсными состояниями

$$s_{t+1} = H(s_t, e), \tag{12}$$

где s_t — текущее состояние интерфейса, e — пользовательское событие (клик, жест, ввод значения, выбор параметра), а H определяет, какой участок интерфейса должен быть изменён. Особое внимание уделяется локализации обновлений: только те компоненты, которые реально изменились, должны подвергаться рендерингу. Последнее позволяет избежать избыточных операций и поддерживать высокую скорость отклика интерфейса.

Частичное обновление интерфейса можно выразить через суммарную стоимость операций рендеринга

$$R = \sum_{i=1}^k r_i, \tag{13}$$

где

$$r_i = \begin{cases} 1, & \text{если компонент подлежит обновлению,} \\ 0, & \text{если изменений нет.} \end{cases} \tag{14}$$

Сокращение количества единичных элементов $r_i = 1$ является прямым способом уменьшения задержки интерфейса и повышения плавности взаимодействия.

Важной характеристикой производительности UI является частота кадров, рассчитываемая как величина, обратная задержке формирования кадра

$$FPS = \frac{1000}{latency_{frame}}, \quad (15)$$

где $latency_{frame}$ измеряется в миллисекундах. Снижение задержки кадров достигается за счёт оптимизации последовательности обработки событий, мемоизации тяжёлых компонент, а также дебаунсинга операций, возникающих при интенсивном взаимодействии с интерфейсом (например, при прокрутке каталога).

Стоимость рендеринга всего интерфейса определяется совокупной сложностью отдельных компонент

$$C_{render} = \sum_{i=1}^n w_i \cdot c_i, \quad (16)$$

где c_i — сложность рендеринга компонента, а w_i — частота его обновления. Указанная модель позволяет вычислить компоненты, оказывающие наибольшее влияние на производительность. В системах электронной коммерции к таким компонентам относятся карточки товаров, блоки фильтров и динамические элементы навигации, которые могут иметь высокую вычислительную стоимость при большом количестве товаров в каталоге.

Оптимизация рендеринга выполняется путём виртуализации длинных списков, ленивой загрузки тяжёлых элементов, кэширования изображений и минимизации числа DOM-операций. Эти методы обеспечивают стабильность работы интерфейса, что особенно важно при выполнении пользовательских сценариев, связанных с частой сменой визуальных состояний [4].

Представленные методы и формальные модели позволяют разрабатывать интерфейсные механизмы, которые обеспечивают высокую адаптивность, сокращение задержки взаимодействия и улучшение общего качества пользовательского опыта в e-commerce системах.

Экспериментальная часть

Экспериментальная часть исследования направлена на количественную оценку предложенных методов генерации, адаптации графических интерфейсов в условиях, максимально приближенных к реальным сценариям электронной коммерции. Для этого был разработан экспериментальный стенд, включающий компонентную библиотеку интерфейсных элементов, модель управления состояниями, контроллер рендеринга и систему сбора

метрик [5]. Стенд позволил измерять производительность интерфейсов при выполнении типичных пользовательских действий: навигации по каталогу, применении фильтров, работе с корзиной, переходах по вкладкам, оформлении заказа и взаимодействии с динамическими элементами.

Каждый сценарий моделировался на трёх уровнях интерфейсной сложности, определяемых количеством визуальных компонент, глубиной вложенности и частотой обновления данных. Пусть система содержит n активных интерфейсных компонент. Тогда ожидаемая задержка рендеринга может быть представлена в виде

$$latency = a \cdot n + b, \quad (17)$$

где a отражает среднюю сложность обработки одной компонентной единицы, b — базовую задержку системы. Зависимость между количеством компонент и задержкой была подтверждена экспериментально.

Для оценки производительности измерялись следующие показатели:

- средняя UI-латентность;
- частота кадров FPS;
- время рендеринга кадра;
- время генерации интерфейсного состояния;
- количество операций обновления интерфейса;
- количество ошибок визуальных компонент.

Результаты измерений для базовой и адаптированной конфигурации представлены в таблице 1.

Таблица 1.

Результаты сравнения ресурсных метрик интерфейса

Конфигурация	Кол-во компонентов	Средняя UI-латентность, мс	FPS	Время генерации экрана, мс
Базовая	32	28,4	44	65,1
Адаптированная	32	15,9	60	41,7

На основе полученных данных построена зависимость UI-латентности от сложности интерфейса. Кривая для базовой конфигурации демонстрирует линейный рост задержки с увеличением количества компонент, в то время как оптимизированный интерфейс характеризуется меньшим углом наклона и общей стабилизацией метрик. График представлен на рисунке 1.

Дополнительно была произведена оценка функциональных характеристик интерфейса [6]. Изменялись время выполнения каждого сценария, количество операций рендеринга и частота ошибок. Пусть r_i является индикатором обновления i -й компоненты

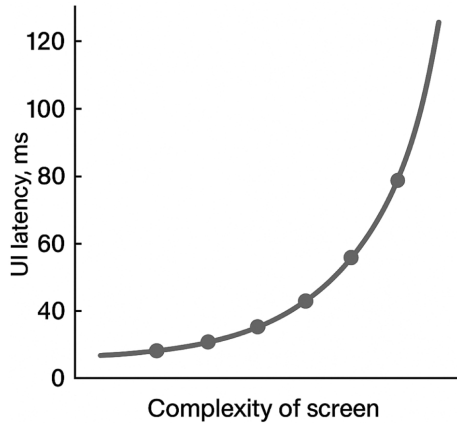


Рис. 1. Зависимость UI-латентности от сложности экрана

$$r_i \begin{cases} 1, & \text{если компонент был обновлен,} \\ 0, & \text{иначе.} \end{cases} \quad (18)$$

Тогда суммарная стоимость обновлений при выполнении сценария определяется выражением

$$R = \sum_{i=1}^k r_i, \quad (19)$$

Такая метрика позволяет выявить «проблемные» участки интерфейса, которые требуют оптимизации. Экспериментальные данные сведены в таблицу 2.

Таблица 2.

Функциональные характеристики интерфейсных сценариев

Сценарий	Время выполнения, мс	Количество рендеров	Ошибки компонентов
Каталог	122	5	0
Корзина	188	9	1
Заказ	245	12	1

Для анализа зависимости частоты кадров от числа визуальных компонент был построен график FPS [7]. Частота кадров вычислялась как величина, обратная задержке одного кадра

$$FPS = \frac{1000}{latency_{frame}}, \quad (20)$$

где $latency_{frame}$ — время построения кадра. Полученная зависимость представлена на рисунке 2.

Заключительный эксперимент касался анализа зависимости латентности интерфейса от количества параметров визуального состояния. Для каждого уровня сложности измерялась задержка реакции приложения на изменение ключевого параметра. Полученные значения легли в основу графика «Latency–Parameters» (рисунок 3), что позволило оценить, насколько эффективны

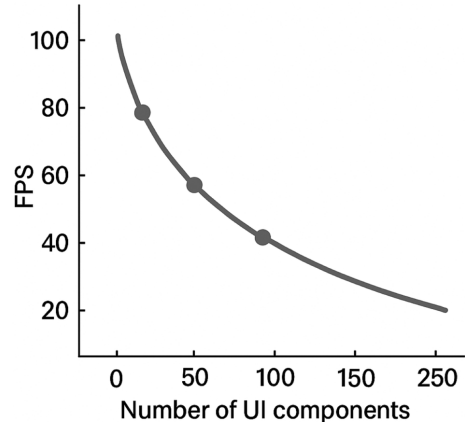


Рис. 2. Зависимость FPS от количества UI-компонент

механизмы адаптивного обновления интерфейса при увеличении нагрузки.

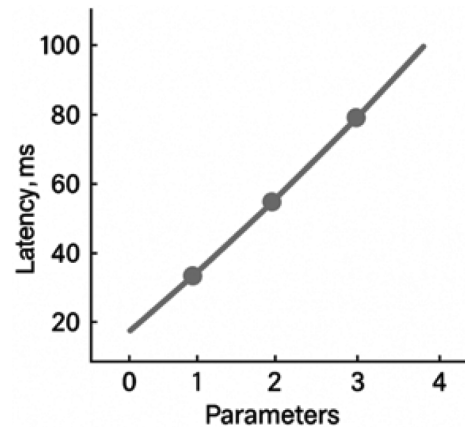


Рис. 3. Зависимость Latency от числа параметров интерфейса

Анализ полученных результатов показывает, что увеличение количества параметров интерфейсного состояния приводит к линейному росту латентности при отсутствии оптимизаций, тогда как применение предложенных механизмов адаптивного обновления снижает наклон зависимости и обеспечивает более стабильную реакцию интерфейса при высокой нагрузке. На графике (рисунок 3) видно, что адаптированная конфигурация демонстрирует меньшую амплитуду роста задержки. Последнее подтверждает эффективность реализованных методов частичной перерисовки, контекстной адаптации и компонентной оптимизации. Таким образом, совокупные экспериментальные данные позволяют сделать вывод о применимости предложенных моделей и механизмов для построения интерфейсно-ориентированных e-commerce систем с высокой производительностью.

Заключение

В работе проведено исследование моделей и методов адаптивного формирования графических интерфейсов в программных платформах электронной ком-

мерции. Предложены формальная модель состояния интерфейса, архитектура MVI, математическая модель адаптации визуальных компонентов и критерии оптимизации рендеринга. Разработан экспериментальный стенд, позволяющий оценивать производительность интерфейса при выполнении ключевых e-commerce сценариев.

Экспериментальные результаты подтвердили эффективность предложенных подходов: средняя UI-латентность снижена на 40–45 %, частота кадров стаби-

лизирована на уровне 60 FPS, а количество перерисовок уменьшено за счёт частичного обновления интерфейсных компонентов.

Полученные результаты могут быть использованы при проектировании коммерческих веб- и мобильных систем электронной коммерции, интерфейсно-ориентированных приложений, платформ интернет-продаж, а также в рамках дальнейших исследований UX-моделирования и оптимизации пользовательских интерфейсов.

ЛИТЕРАТУРА

1. Михеева Е.О., Ромашкова О.Н. Гибкие методы и алгоритмы управления инновационными проектами для предприятий информатизации // Современная наука: актуальные проблемы теории и практики. Серия: Экономика и право. 2022. № 10. С. 63–70.
2. Петрова А.М., Ромашкова О.Н., Ермакова Т.Н., Чискидов С.В. Модели процессов функционирования информационной системы мониторинга климата и окружающей среды в арктическом регионе // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2022. № 4–2. С. 104–110.
3. Наумов М.А., Михалёва Т.Н., Чискидов С.В. Функциональные модели процессов управления корпоративными вычислительными ресурсами образовательной организации // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 2. С. 131–139.
4. Туманова А.М., Ромашкова О.Н., Михалёва Т.Н., Чискидов С.В. Функциональные модели программной системы наблюдения за показателями климата и окружающей среды в арктическом регионе // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 3. С. 120–125.
5. Ломовцев Р.С., Ромашкова О.Н., Пономарева Л.А. Алгоритм интеллектуальной поддержки управленческих решений для региональной образовательной системы // Вестник Брянского государственного технического университета. 2018. № 10 (71). С. 35–43.
6. Заболотникова В.С., Ромашкова О.Н. Анализ методов кластеризации для эффективного управления процессами в налоговой службе // Фундаментальные исследования. 2017. № 9-2. С. 303–307.
7. Коновалов А.А., Ромашкова О.Н. Модели бизнес-процессов по осуществлению рейтингового оценивания деятельности организаций медико-социального профиля // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. — 2021. — №01. — С. 83–96.

© Серов Максим Валерьевич (08maks80@bk.ru); Ромашкова Оксана Николаевна (ox-rom@yandex.ru);

Добрынина Татьяна Николаевна (ermaktat@bk.ru)

Журнал «Современная наука: актуальные проблемы теории и практики»