Т. 12. №3 2026
https://doi.org/10.33619/2414-2948/124

УДК 004.428.4 https://doi.org/10.33619/2414-2948/124/13

ИСПОЛЬЗОВАНИЕ REACT ДЛЯ СОЗДАНИЯ ИНТЕРАКТИВНЫХ ЭЛЕМЕНТОВ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

©Сафронов А. М., *МИРЭА — Российский технологический университет (РТУ МИРЭА), г. Москва, Россия, sanchellos27@mail.ru*
©Зеленова Ю. И., *ORCID: 0000-0002-6979-2443, SPIN-код: 4568-0055, канд. техн. наук, МИРЭА — Российский технологический университет (РТУ МИРЭА), г. Москва, Россия, zelenova.julie@yandex.ru*

USING REACT FOR CREATING INTERACTIVE USER INTERFACE ELEMENTS

©Safronov A., *Russian Technological University – MIREA (RTU MIREA), Moscow, Russia, sanchellos27@mail.ru*
©Zelenova Ju., *ORCID: 0000-0002-6979-2443, SPIN-code: 4568-0055, Ph.D., Russian Technological University – MIREA (RTU MIREA), Moscow, Russia, zelenova.julie@yandex.ru*

*Аннотация***.** Статья посвящена комплексному анализу использования библиотеки React в процессе создания интерактивных элементов пользовательского интерфейса в современных веб-приложениях. В центре внимания находится архитектурная логика React, рассматриваемая как основа формирования устойчивых, масштабируемых и предсказуемых интерфейсных решений. Интерактивность трактуется не как совокупность визуальных эффектов, а как системное свойство интерфейса, определяемое взаимосвязью состояния, компонентной структуры и событийной модели. В работе последовательно раскрываются принципы декларативного описания интерфейса, компонентного подхода и однонаправленного потока данных, которые определяют специфику проектирования интерактивных элементов в React. Особое внимание уделяется роли состояния как ключевого механизма управления поведением интерфейса, а также разграничению локального и глобального состояния в контексте сложных пользовательских сценариев. Анализируется значение виртуального представления интерфейса как инструмента оптимизации обновлений и обеспечения согласованности визуального отображения при частых изменениях данных. Использование виртуального DOM позволяет минимизировать количество операций с реальной структурой документа, что имеет принципиальное значение для производительности интерактивных интерфейсов. Рассматривается событийная модель React как формализованный механизм связи пользовательских действий с изменениями состояния, обеспечивающий детерминированность и воспроизводимость поведения интерфейсных элементов. В рамках сравнительного анализа выявляются отличия React-подхода от традиционных императивных моделей построения пользовательского интерфейса, что позволяет обосновать его архитектурные преимущества при разработке сложных интерфейсных систем. Таким образом, React следует рассматривать не только как инструмент прикладной разработки, но и как методологическую основу проектирования интерактивных пользовательских интерфейсов, ориентированную на логическую связность, управляемость и долгосрочную устойчивость программных решений.

Abstract. The article provides a comprehensive analysis of the use of the React library in the creation of interactive user interface elements within modern web applications. The focus is placed on the architectural logic of React as a foundational framework for building stable, scalable, and predictable interface solutions. Interactivity is interpreted not as a set of isolated visual effects but as

Т. 12. №3 2026
https://doi.org/10.33619/2414-2948/124

a systemic property of the interface, determined by the relationship between state management, component structure, and the event-handling model. The paper consistently examines the principles of declarative interface description, component-based architecture, and unidirectional data flow, which define the specific features of designing interactive elements in React. Particular attention is given to the role of state as the central mechanism governing interface behavior, as well as to the distinction between local and global state in the context of complex user interaction scenarios. The significance of the virtual representation of the interface is analyzed as a means of optimizing updates and maintaining visual consistency under conditions of frequent data changes. The article demonstrates that the use of a virtual DOM makes it possible to reduce direct manipulations of the actual document structure, which is critically important for the performance and stability of interactive interfaces. The event model implemented in React is examined as a formalized mechanism linking user actions to state transitions, ensuring deterministic and reproducible behavior of interface components. A comparative analysis highlights the differences between the React approach and traditional imperative models of user interface development, allowing for a systematic justification of its architectural advantages in the design of complex interface systems. The study concludes that React should be regarded not merely as a practical development tool, but as a methodological foundation for designing interactive user interfaces oriented toward logical consistency, controllability, and long-term architectural sustainability of software solutions.

Ключевые слова: компонентная архитектура, декларативный интерфейс, React, виртуальный DOM, событийная модель, интерфейсная логика.

Keywords: component architecture, declarative interface, React, virtual DOM, event model, interface logic.

Развитие веб-приложений в последнее время характеризуется смещением акцента от статических интерфейсов к динамическим пользовательским средам, в которых ключевую роль играет интерактивность, адаптивность и высокая отзывчивость интерфейса на действия пользователя. Пользовательский интерфейс перестал рассматриваться исключительно как визуальная оболочка программного продукта и приобрёл статус полноценного функционального слоя, напрямую влияющего на эффективность взаимодействия человека с информационными системами, на когнитивную нагрузку, а также на производственные и коммерческие показатели цифровых сервисов. В этих условиях особую значимость приобретают программные инструменты, позволяющие формировать сложные интерфейсные структуры с предсказуемым поведением, масштабируемой архитектурой и строгой логикой управления состоянием [1].

Одним из наиболее распространённых инструментов такого рода является библиотека React, ориентированная на компонентный подход к построению пользовательских интерфейсов. Концептуальная основа React связана с представлением интерфейса как функции от состояния приложения, что принципиально отличает данный подход от традиционных императивных моделей манипуляции элементами DOM. Использование декларативного описания интерфейса позволяет формализовать логику взаимодействия компонентов, снизить связность между отдельными частями пользовательского интерфейса и обеспечить более высокий уровень управляемости сложных интерфейсных сценариев. В результате React применяется не только в клиентских веб-приложениях, но и в системах с повышенными требованиями к устойчивости интерфейсной логики, включая корпоративные платформы, образовательные цифровые среды и высоконагруженные сервисы.

Т. 12. №3 2026
https://doi.org/10.33619/2414-2948/124

Интерактивные элементы пользовательского интерфейса в контексте React представляют собой не изолированные визуальные компоненты, а структурные единицы, инкапсулирующие как представление, так и поведенческую логику. К таким элементам относятся формы с валидацией, динамические списки, панели навигации, модальные окна, элементы визуализации данных и другие интерфейсные конструкции, требующие точного управления состоянием и событийной моделью. Особенность React заключается в том, что интерактивность достигается не за счёт прямого вмешательства в структуру документа, а посредством согласованного обновления виртуального представления интерфейса, что обеспечивает предсказуемость изменений и упрощает анализ поведения системы.

Актуальность исследования использования React для создания интерактивных элементов пользовательского интерфейса обусловлена необходимостью теоретического осмысления архитектурных и логических принципов, лежащих в основе данной технологии, а также их влияния на качество программных решений. Несмотря на широкое практическое применение React, в научных публикациях нередко преобладает описательный или прикладной подход, в то время как вопросы формальной организации компонентной структуры, управления состоянием, оптимизации перерисовки интерфейса и обеспечения согласованности пользовательских сценариев требуют более глубокого аналитического рассмотрения. Особенно важным представляется анализ того, каким образом архитектурные решения React соотносятся с задачами построения интерактивных интерфейсов в условиях усложнения пользовательских требований и роста функциональной насыщенности веб-приложений.

Целью настоящего исследования является комплексное рассмотрение использования React как инструмента создания интерактивных элементов пользовательского интерфейса с позиций архитектурной логики, модели управления состоянием и механизмов обновления представления. В рамках исследования предполагается выявить ключевые принципы, определяющие эффективность применения React для реализации интерактивности, а также показать, каким образом данные принципы влияют на структурную целостность интерфейса и устойчивость его поведения. Реализация поставленной цели предполагает последовательный анализ концептуальных оснований React, особенностей компонентного подхода и специфики взаимодействия между пользовательскими событиями и состоянием интерфейса.

Научная новизна работы заключается в систематизированном рассмотрении React не как совокупности практических приёмов разработки, а как целостной архитектурной модели построения интерактивных пользовательских интерфейсов. Такой подход позволяет перейти от фрагментарного описания отдельных механизмов к формированию связной теоретической картины, отражающей внутреннюю логику работы React и её значение для проектирования современных веб-приложений.

Использование React для создания интерактивных элементов пользовательского интерфейса базируется на принципиально ином представлении интерфейса как программной абстракции. В рамках данной библиотеки пользовательский интерфейс интерпретируется не как совокупность разрозненных DOM-элементов, а как иерархия компонентов, каждый из которых описывает собственное состояние и правила визуального отображения. Такое представление формирует целостную модель интерфейсной системы, в которой логика взаимодействия и визуальное представление не существуют изолированно, а образуют функционально связанную структуру.

Компонент в React выступает как минимальная единица интерфейсной архитектуры, обладающая строго определёнными границами ответственности. Он инкапсулирует структуру разметки, стили отображения и поведенческую логику, связанную с пользовательскими событиями. Данный подход позволяет формировать интерфейсы с высокой степенью

Т. 12. №3 2026
https://doi.org/10.33619/2414-2948/124

модульности, что принципиально важно при разработке сложных интерактивных систем. В отличие от традиционных подходов, основанных на прямой манипуляции DOM, компонентная модель React исключает неявные зависимости между элементами интерфейса и снижает вероятность неконтролируемых побочных эффектов [2].

Архитектурное значение React также проявляется в отказе от императивного управления пользовательским интерфейсом. Интерфейсное состояние в React задаётся декларативно, что означает, что разработчик описывает не последовательность операций изменения интерфейса, а условия, при которых интерфейс должен находиться в определённом визуальном состоянии. Такое описание существенно упрощает формализацию интерактивных сценариев и повышает воспроизводимость поведения интерфейса при различных пользовательских действиях. Ключевым элементом интерактивных пользовательских интерфейсов, создаваемых с использованием React, является управление состоянием. Состояние в данном контексте представляет собой совокупность данных, определяющих текущее отображение и поведение интерфейсных элементов. Любое пользовательское действие, включая ввод данных, навигацию, активацию элементов управления или изменение параметров отображения, приводит к трансформации состояния, что, в свою очередь, инициирует обновление интерфейса.

React реализует строгую модель однонаправленного потока данных, при которой изменения состояния распространяются сверху вниз по иерархии компонентов. Такой подход обеспечивает предсказуемость поведения интерфейса и позволяет однозначно определить источник каждого изменения. В условиях высокой интерактивности данная модель приобретает особую значимость, поскольку исключает циклические зависимости и неконтролируемые обновления интерфейсных элементов. Интерактивные элементы, такие как формы, выпадающие списки, переключатели состояний и элементы визуализации данных, в React реализуются как управляемые компоненты. Их состояние хранится либо локально внутри компонента, либо во внешнем хранилище состояния, в зависимости от масштабов интерфейсной логики. При этом каждое изменение пользовательского ввода немедленно отражается в состоянии компонента, а визуальное представление автоматически синхронизируется с актуальными данными. Такая модель обеспечивает логическую согласованность интерфейса и минимизирует расхождения между пользовательскими действиями и отображаемым результатом. Особое значение в контексте интерактивности имеет разграничение локального и глобального состояния. Локальное состояние используется для управления поведением отдельных компонентов, тогда как глобальное состояние применяется для синхронизации данных между различными частями интерфейса. React допускает интеграцию специализированных механизмов управления состоянием, что позволяет масштабировать интерфейсные решения без нарушения архитектурной целостности. Одной из фундаментальных особенностей React является использование виртуального представления интерфейса, которое служит промежуточным слоем между логикой приложения и реальным DOM. Виртуальное представление представляет собой абстрактное дерево элементов, отражающее текущее состояние интерфейса. При изменении состояния React формирует новое виртуальное дерево и выполняет его сопоставление с предыдущей версией, определяя минимальный набор изменений, необходимых для обновления реального интерфейса [3].

Данный механизм имеет принципиальное значение для создания интерактивных элементов, поскольку позволяет эффективно обрабатывать частые изменения состояния, характерные для пользовательского взаимодействия. В условиях интенсивной интерактивности, когда пользовательские действия могут приводить к множественным

Т. 12. №3 2026
https://doi.org/10.33619/2414-2948/124

обновлениям интерфейса в короткий промежуток времени, оптимизация обновлений становится критически важной. React обеспечивает такую оптимизацию за счёт минимизации операций с реальным DOM, которые являются наиболее ресурсоёмкими. Использование виртуального представления также способствует формированию устойчивых интерфейсных сценариев. Разработчик может оперировать логическими состояниями и абстрактными компонентами, не вникая в детали низкоуровневого управления элементами страницы. Это снижает сложность проектирования интерактивных интерфейсов и позволяет сосредоточиться на логике пользовательского взаимодействия. Интерактивность пользовательского интерфейса в React реализуется через событийную модель, которая обеспечивает связь между пользовательскими действиями и изменениями состояния. События в React обрабатываются на уровне компонентов и представляют собой абстракцию над нативными событиями браузера. Такой подход обеспечивает унифицированный механизм обработки пользовательского ввода независимо от платформенных особенностей. Каждый интерактивный элемент интерфейса, будь то кнопка, поле ввода или сложный составной компонент, реагирует на события посредством функций-обработчиков. Эти функции инициируют изменение состояния, что приводит к обновлению соответствующих компонентов. Важно отметить, что обработка событий в React строго подчинена принципу детерминизма: при одинаковом состоянии и одинаковом событии результат обновления интерфейса будет идентичен. Это свойство имеет принципиальное значение для проектирования надёжных интерактивных систем [4].

Интерактивные элементы в React могут быть как атомарными, так и составными. Составные компоненты объединяют несколько элементов управления и реализуют сложные сценарии взаимодействия, включая пошаговые формы, динамическую фильтрацию данных и адаптивную навигацию. При этом каждый уровень вложенности компонентов сохраняет собственную логику обработки событий, что позволяет выстраивать многоуровневые интерфейсные структуры без потери управляемости. Для более наглядного понимания особенностей использования React целесообразно сопоставить его архитектурные принципы с традиционными подходами к созданию интерактивных пользовательских интерфейсов. Такое сопоставление позволяет выявить системные преимущества компонентной и декларативной модели React в контексте управления интерактивностью (Таблица).

Таблица

СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ПОДХОДОВ
К СОЗДАНИЮ ИНТЕРАКТИВНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

Критерий анализа	*Традиционные подходы к UI*	*React-подход*
Модель управления интерфейсом	Императивная, основанная на прямых изменениях DOM	Декларативная, основанная на состоянии
Организация интерактивных элементов	Разрозненные обработчики событий	Компонентная инкапсуляция логики
Управление состоянием	Неявное, распределённое	Формализованное, однонаправленное
Масштабируемость интерфейса	Ограниченная при росте сложности	Высокая за счёт модульности
Предсказуемость поведения	Зависит от порядка операций	Определяется состоянием и входными данными

Представленные различия демонстрируют, что использование React позволяет перейти от фрагментарного управления интерактивностью к системному проектированию пользовательских интерфейсов. Компонентная структура и декларативное описание

Т. 12. №3 2026
https://doi.org/10.33619/2414-2948/124

интерфейса формируют основу для устойчивых и масштабируемых решений, особенно в условиях усложнения пользовательских сценариев. Использование React в современных веб-приложениях отражает более широкий сдвиг в парадигме проектирования пользовательских интерфейсов. Интерактивность перестаёт быть набором отдельных эффектов и превращается в системное свойство интерфейса, встроенное в его архитектуру. React предоставляет инструментарий для формализации этой интерактивности, обеспечивая строгую связь между состоянием, событиями и визуальным представлением [5].

В условиях роста функциональной насыщенности цифровых сервисов и увеличения требований к пользовательскому опыту React выступает как средство обеспечения архитектурной устойчивости интерфейсных решений. Его применение позволяет разрабатывать интерфейсы, в которых интерактивные элементы не нарушают целостность системы, а, напротив, усиливают её логическую связанность. Таким образом, использование React для создания интерактивных элементов пользовательского интерфейса следует рассматривать не только как технологический выбор, но и как методологическое решение, определяющее качество и надёжность современных веб-приложений.

Проведённый анализ позволяет сделать вывод о том, что использование React для создания интерактивных элементов пользовательского интерфейса представляет собой не частное технологическое решение, а целостный архитектурный подход к проектированию современных веб-приложений. React формирует концептуальную модель, в рамках которой интерактивность интерфейса изначально заложена в структуру приложения и определяется взаимосвязью состояния, компонентов и событийной логики. Такой подход обеспечивает системность построения пользовательских интерфейсов и позволяет отказаться от фрагментарного управления визуальными элементами. Ключевое значение React заключается в формализации процесса управления состоянием как основного механизма интерактивности. Чёткое разграничение источников данных, однонаправленный поток изменений и детерминированная реакция интерфейса на пользовательские действия создают условия для предсказуемого и воспроизводимого поведения интерфейсных элементов. Это обстоятельство приобретает особую значимость при разработке сложных интерфейсов, в которых множественные пользовательские сценарии требуют строгой логической согласованности и устойчивости к изменению контекста взаимодействия.

Использование виртуального представления интерфейса и механизмов оптимизации обновлений позволяет рассматривать React как инструмент, обеспечивающий баланс между функциональной насыщенностью интерфейса и его производительностью. Интерактивные элементы, характеризующиеся частыми изменениями состояния, реализуются без прямого вмешательства в структуру документа, что снижает вычислительные издержки и повышает стабильность работы интерфейса. В результате достигается высокий уровень адаптивности пользовательского интерфейса без утраты управляемости архитектуры приложения. Компонентная организация интерфейса в React способствует формированию устойчивых и масштабируемых решений, в которых интерактивные элементы не изолированы, а встроены в иерархическую структуру приложения. Инкапсуляция логики и визуального представления внутри компонентов позволяет сохранять целостность интерфейса при его расширении и модификации, что особенно важно в условиях долгосрочной эксплуатации программных продуктов и эволюции пользовательских требований. В совокупности изложенные положения позволяют утверждать, что React представляет собой эффективную архитектурную основу для создания интерактивных пользовательских интерфейсов, ориентированную на логическую связность, предсказуемость поведения и структурную устойчивость. Применение данного подхода обеспечивает переход от эмпирического конструирования интерфейсов к

Т. 12. №3 2026
https://doi.org/10.33619/2414-2948/124

формализованному проектированию интерактивных систем, что соответствует современным требованиям к качеству и надёжности цифровых продуктов.

Список литературы:

1. Антонов С. А., Вуколов А. А., Кононыхина К. А. Обзор современных библиотек для разработки интерфейса веб приложения // Вестник науки. 2024. Т. 5. №9(78). С. 477-504.

2. Еремейчик К. Ю., Малашенко Д. А. Программное средство учета и планирования использования операционных блоков // Компьютерные системы и сети: материалы 61-й научной конференции. Минск, 2025. С. 114–115.

3. Кравцов Е. П. Разработка высокопроизводительных React-приложений: методы и практики оптимизации // European science. 2024. №1 (69). С. 53-58.

4. Нефёдов Д. С., Шехирев Д. Р. Обоснование выбора react для разработки веб-приложения для тестирования в образовательных организациях: сравнение с нативным javascript // Парадигма. 2025. №4-2. С. 315-321.

5. Семёнов В. В., Лапицкий С. А. Повышение интерактивности веб-страниц: анализ React и Vue // Дневник науки. 2024. №12.

References:

1. Antonov, S. A., Vukolov, A. A., & Kononykhina, K. A. (2024). Obzor sovremennykh bibliotek dlya razrabotki interfeisa veb prilozheniya. *Vestnik nauki, 5*(9 (78)), 477-504. (in Russian).

2. Eremeichik, K. Yu., & Malashenko, D. A. (2025). Programmnoe sredstvo ucheta i planirovaniya ispol'zovaniya operatsionnykh blokov. In *Komp'yuternye sistemy i seti: materialy 61-i nauchnoi konferentsii, Minsk,* 114–115. (in Russian).

3. Kravtsov, E. P. (2024). Razrabotka vysokoproizvoditel'nykh React-prilozhenii: metody i praktiki optimizatsii. *European science,* (1 (69)), 53-58. (in Russian).

4. Nefedov, D. S., & Shekhirev, D. R. (2025). Obosnovanie vybora react dlya razrabotki veb-prilozheniya dlya testirovaniya v obrazovatel'nykh organizatsiyakh: sravnenie s nativnym javascript. *Paradigma,* (4-2), 315-321. (in Russian).

5. Semenov, V. V., & Lapitskii, S. A. (2024). Povyshenie interaktivnosti veb-stranits: analiz React i Vue. *Dnevnik nauki,* (12). (in Russian).

Поступила в редакцию
22.01.2026 г.

Принята к публикации
30.01.2026 г.

Ссылка для цитирования:

Сафронов А. М., Зеленова Ю. И. Использование React для создания интерактивных элементов пользовательского интерфейса // Бюллетень науки и практики. 2026. Т. 12. №3. С. 126-132. https://doi.org/10.33619/2414-2948/124/13

Cite as (APA):

Safronov, A., & Zelenova, Ju. (2026). Using React for Creating Interactive user Interface Elements. *Bulletin of Science and Practice, 12*(3), 126-132. (in Russian). https://doi.org/10.33619/2414-2948/124/13