

Министерство образования Республики Беларусь

Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.58

Журавлева
Марина Витальевна

МОДЕЛЕОРИЕНТИРОВАННЫЕ СРЕДСТВА ПОСТРОЕНИЯ
ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ВЕБ-ПРИЛОЖЕНИЙ

АВТОРЕФЕРАТ

диссертации на соискание степени магистра технических наук
по специальности 1-31 80 10 «Теоретические основы информатики»

Научный руководитель

Шункевич Даниил Вячеславович

Заведующий кафедрой ИИТ, канди-
дат технических наук

Минск 2020

КРАТКОЕ ВВЕДЕНИЕ

На текущий момент существует множество решений в области построения модели пользовательских интерфейсов. Разработано множество подходов и выделены различные уровни абстракции. Однако взгляды на разработку пользовательских интерфейсов меняются каждые полгода. Это говорит о том, что каждый старый подход обладал рядом недостатков, из-за которых разработчики были вынуждены использовать новое решение. В то же время наличие большого количества различных подходов к разработке затрудняет использование уже созданных модулей и компонентов.

Таким образом, сообщество разработчиков приходит к тому состоянию, что в будущем будет невозможно использовать ничего из того огромного количества наработок, которое есть сегодня. Для разработчика изучать все новые и новые подходы становится затруднительно, так как «порог вхождения» в новые технологии постоянно повышается, а времени на изучение становится все меньше. Разработчики не успевают в совершенстве овладеть технологией до того, как она устаревает и становится невостребованной. Из-за этого повышается количество некачественного кода и некачественных продуктов.

Пользовательский интерфейс является неотъемлемой составляющей подавляющего большинства программных средств. Как правило, пользователи судят о программном средстве в целом и «пригодности» его к использованию по тому, насколько удобен пользовательский интерфейс и в какой мере он отвечает их требованиям. При этом требования пользователей весьма разнообразны и определяются как личными предпочтениями и особыми запросами, так и назначением интерфейсов, особенностями представления информации в различных предметных областях, условиями и средой использования. Эти факторы требуют создания интерфейсов для различных платформ с поддержкой множества типов диалогов (основанных на экранных формах или WIMP-интерфейсах, текстах, графических образах, речи, с использованием различных цветовых схем, анимации и мн. др.). При этом важным критерием является их качество.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность

На текущий момент существует множество решений в области построения моделей пользовательских интерфейсов. Разработано множество подходов и выделены различные уровни абстракции. Однако взгляды на разработку пользовательских интерфейсов меняются каждые полгода. Это говорит о том, что каждый старый подход обладал рядом недостатков, из-за которых разработчики были вынуждены использовать новое решение. В то же время наличие большого количества различных подходов к разработке затрудняет использование уже созданных модулей и компонентов.

Таким образом, сообщество разработчиков приходит к тому состоянию, что в будущем будет невозможно использовать ничего из того огромного количества наработок, которое есть сегодня. Для разработчика изучать все новые и новые подходы становится затруднительно, так как «порог вхождения» в новые технологии постоянно повышается, а времени на изучение становится все меньше. Разработчики не успевают в совершенстве овладеть технологией до того, как она устаревает и становится неостребованной. Из-за этого повышается количество некачественного кода и некачественных продуктов.

В связи с вышесказанным актуальными являются исследования, направленные на решение проблем уменьшения трудоемкости проектирования, прототипирования, реализации и сопровождения пользовательского интерфейса, с поддержкой различных типов диалога. Не менее острой проблемой является изменение требований, выдвигаемых к конечному продукту еще до его реализации. Это вызвано глобальной цифровизацией, бурным ростом пользовательских потребностей и высоким уровнем конкурентности. Разрабатываемый продукт должен соответствовать последним тенденциям в удобстве и потребностях пользователя. Для решения данной проблемы были созданы «Гибкие» методологии разработки, однако они не способны решить основную проблему – необходимость ручного изменения программного кода и адаптация его под новые требования. Поэтому мы можем наблюдать «двойную» трату времени на разработку функционала: изначальная разработка и исправление под новые требования. Зачастую, исправление уже существующего функционала отнимает больше времени, чем создание его по первоначальным требованиям.

Цели и задачи исследования

Целью диссертационной работы является разработка концепции, моделей, методов и инструментария для создания, реализации и сопровождения пользовательских интерфейсов.

Для достижения указанной цели в работе используются метод автоматической кодогенерации элементов пользовательского интерфейса, а также моделиориентированный подход к разработке интерфейсов. В соответствии с целью диссертационной работы поставлены следующие *задачи исследования*.

а) Разработка концепции автоматизации проектирования, реализации, сопровождения и изменения пользовательского интерфейса на основе моделиориентированного подхода.

б) Разработка моделей пользовательского интерфейса для каждой группы задач.

в) Разработка алгоритмов генерации программного кода из моделей пользовательского интерфейса.

г) Разработка методов обеспечения обратной совместимости генерируемого программного кода для большего пользовательского охвата, путем поддержки старых браузеров.

д) Разработка методов обеспечения адаптивности созданных моделей, для поддержки мобильных устройств и планшетов.

е) Описание вариантов пользовательского взаимодействия с системой.

ж) Тестирование и отладка технологии для проектирования и создания пользовательских интерфейсов.

Структура и объём диссертации

Диссертация изложена на 64 страницах. Она состоит из перечня условных обозначений (1 страница), введения (2 страницы), анализа существующих подходов (20 страниц), разработки архитектуры и описания алгоритмов (13 страниц), реализации приложения (11 страниц), практического применения и сравнения полученного приложения с существующими аналогами (12 страниц), заключения (1 страница), списка использованных источников, состоящий из 23 наименований (2 страницы). Работа содержит 20 иллюстраций и две таблицы.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе был проведен анализ литературы и существующих средств разработки пользовательских веб-интерфесов, который выявил основные достижения, а также существенные проблемы разработки (проектирования, реализации, сопровождения) пользовательских интерфейсов, их оценивания и создания систем помощи. Также была предложена архитектура, которая решает основные проблемы моделиориентированного подхода:

- предлагаемый подход решает проблему порога вхождения путем использования самого распространенного языка веб-разработки, средств и методов разработки, а также хорошо структурированного исходного программного кода и программной документации;

- для разработки данного программного продукта будет применяться модельный (или компонентный) подход, что позволит элементам интерфейса быть независимыми друг от друга, легко их взаимозаменять;

- будет решена проблема обратной совместимости путем использования только проверенных технологий с максимальным охватом, а также механизмов, которые обеспечивают преобразование нового кода в код с более широкоисовместимостью.

Во второй главе была описана концепция применения моделиориентированного подхода в разработке пользовательского веб-интерфейса и архитектуры генератора интерфейсов на его основе. Предлагаемая архитектура базируется на моделиориентированном (или компонентном) подходе. Особенность данной архитектуры в универсальности моделей. Это позволяет добиться повторного их использования путем минимальной кастомизации во всем интерфейсе.

Основным хранилищем для моделей в системе является ядро. Ядро содержит в себе набор базовых моделей, с помощью которых строится интерфейс. Каждая модель представляет собой независимый элемент интерфейса, решающий определенную и конкретную задачу.

Каждая модель представляет собой независимый элемент интерфейса, решающий определенную и конкретную задачу. Это основной строительный блок интерфейса. В теории пользовательского опыта базовые элементы интерфейса делятся на 2 вида: элементы представления и элементы взаимодействия. *Модели представления* необходимы для отображения данных. Пользователь не может с ними взаимодействовать. При помощи моделей отображения пользователь получает информацию из системы, а система общается с пользователем соответственно. *Интерактивные модели* представляют собой основной элемент пользовательского взаимодействия. С их помощью пользователь общается с системой и влияет на нее. Данные модели позволяют преобразовывать пользовательские данные в информа-

цию приложения и оказывать влияние как на систему в целом, так и на отдельные модели представления. Каждая такая модель имеет входные и выходные данные.

В третьей главе было представлено обоснование выбранных средств разработки для приложения по разработке веб-интерфейсов. Алгоритм генерации моделей интерфейса, как и весь программный код написаны с использованием самого популярного языка для веб разработки - JavaScript. Для поддержки хорошей структурированности и читабельности исходного программного кода используется механизм статического анализа кода при помощи линтера eslint. Это самый популярный в мире статический анализатор кода, написанного на JavaScript. Данный механизм можно гибко настроить под структуру и стилистику конкретного проекта. Он хорошо интегрируется с редакторами кода и системами сборки а также регулярно обновляется, чтобы поддерживать последние стандарты. Также eslint поддерживает автоматическое исправление кода.

Для написания удобной и хорошо структурированной программной документации применяется JSDOC. Это генератор документации в HTML-формате из комментариев исходного кода на JavaScript. Данный механизм позволяет писать документацию к программному коду прямо внутри него и генерировать документацию в нужном формате, HTML, PDF и т.д. Данный механизм зарекомендовал себя в таких распространенных библиотеках как React, Lodash, Ramda, ReactBootstrap и др.

Для того, чтобы использовать только проверенные технологии с максимальным пользовательским охватом, каждая выбранная технология подвергается проверке при помощи информации из CanIUse. Это самая обширная база данных о поддержке технологий.

Преобразование исходного кода в код с более широкой совместимостью обеспечивается с помощью Webpack. Это статический модульный сборщик для приложений на JavaScript, который также позволяет осуществлять преобразование программного кода с помощью установки отдельных модулей и плагинов. Он позволяет преобразовывать синтаксис из новых стандартов в более поддерживаемый.

Данные технологии позволяют построить наиболее качественное приложение и соответствовать самым современным стандартам разработки ПО.

В четвертой главе продемонстрирован пример работы приложения для построения пользовательских интерфейсов. Были проиллюстрированы основные этапы взаимодействия с приложением:

- авторизация и регистрация
- создание прототипа веб-интерфейса
- примеры добавления интерактивных элементов
- примеры добавления визуальных элементов

- примеры копирования и удаления элементов
- внешний вид приложения на мобильных устройствах
- пример и структура сгенерированного программного кода.

Также было проведено сравнение приложения для построения пользовательских интерфейсов с существующими аналогами и показаны основные преимущества данного приложения.

В качестве будущих возможных улучшений может быть совместная работа над созданием интерфейса нескольких человек, а также импортирование макетов из таких инструментов прототипирования, как photoshop, sketch, zeplin.

Библиотека БГУИР

ЗАКЛЮЧЕНИЕ

В рамках магистерской диссертации были рассмотрены существующие сервисы для создания веб-интерфейсов. На основе проведенного анализа литературы и существующих средств разработки пользовательских веб-интерфейсов были выявлены основные достижения, а также существенные проблемы разработки (проектирования, реализации, сопровождения).

На основе выделенных проблем был предложен подход и решение трех основных проблем. Далее была подробно описана архитектура, которая базируется на модели ориентированном подходе. Также был описан основной алгоритм системы и взаимодействия компонентов в системе. В свою очередь, каждый компонент системы был подробно описан, указана его роль в системе и способ взаимодействия с другими компонентами.

Затем были приведены и подробно обоснованы выбранные средства и технологии разработки, а также построены диаграммы деятельности для детального описания алгоритмов системы.

Далее в работе были продемонстрированы результаты работы алгоритмов системы, был произведен обзор и анализ данных результатов. В результате анализа были выявлены сильные и слабые стороны системы, а также, предложен ряд способов улучшений и оптимизаций системы.

В итоге разработанный подход к построению системы для разработки модели ориентированных веб-интерфейсов имеет ряд преимуществ:

- а) удобное создание макета веб-интерфейса;
- б) автоматическая генерация кода в React;
- в) полученный интерфейс адаптивен под любые устройства;
- г) возможность интеграции в другой интерфейс приложения;
- д) поддержка SPA режима;
- е) расширяемость моделей;
- ж) возможность повторного использования моделей;
- з) поддержка aria-атрибутов.

В качестве будущих возможных улучшений может быть совместная работа над созданием интерфейса нескольких человек, а также импортирование макетов из таких инструментов прототипирования, как photoshop, sketch, zeplin.

В результате цель магистерской диссертации была достигнута.