УНИВЕРСИТЕТ ИТМО

1

Альманах

НАУЧНЫХ РАБОТ МОЛОДЫХ УЧЕНЫХ

2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

АЛЬМАНАХ НАУЧНЫХ РАБОТ МОЛОДЫХ УЧЕНЫХ Университета ИТМО

Том 1

Санкт-Петербург

2019

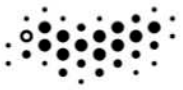
Альманах научных работ молодых ученых Университета ИТМО. Том 1. – СПб.: Университет ИТМО, 2019. – 324 с.

Издание содержит результаты научных работ молодых ученых, доложенные на XLVIII научной и учебно-методической конференции Университета ИТМО по тематике: программная инженерия и компьютерная техника.

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Председатель редколлегии:
Кустарев Павел Валерьевич
кандидат технических наук, декан факультета программной инженерии и компьютерной техники
Члены редколлегии:
Алиев Тауфик Измайлович
доктор технических наук, профессор факультета программной инженерии и компьютерной техники
Быковский Сергей Вячеславович
кандидат технических наук, доцент факультета программной инженерии и компьютерной техники
Белозубов Александр Владимирович
кандидат технических наук, преподаватель факультета программной инженерии и компьютерной техники
Болдырева Елена Александровна
аспирант факультета программной инженерии и компьютерной техники

ISBN 978-5-7577-0606-1
ISBN 978-5-7577-0614-6 (Том 1)



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2019
© Авторы, 2019

ВВЕДЕНИЕ

Издание содержит результаты научных работ молодых ученых, доложенные 29 января – 1 февраля 2019 года на XLVIII научной и учебно-методической конференции Университета ИТМО по тематике: программная инженерия и компьютерная техника.

Конференция проводится в целях усиления интегрирующей роли университета в области научных исследований по приоритетным направлениям развития науки, технологий и техники и ознакомления научной общественности с результатами исследований, выполненных в рамках государственного задания Министерства образования и науки РФ, программы развития Университета ИТМО на 2009–2018 годы, программы повышения конкурентоспособности Университета ИТМО среди ведущих мировых научно-образовательных центров на 2013–2020 гг., Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 годы», грантов Президента РФ для поддержки молодых российских ученых и ведущих научных школ, грантов РФФИ, РГНФ, РНФ и Правительства РФ (по постановлению № 220 от 09.04.2010 г.) и по инициативным научно-исследовательским проектам, проводимым учеными, преподавателями, научными сотрудниками, аспирантами, магистрантами и студентами университета, в том числе в содружестве с предприятиями и организациями Санкт-Петербурга, а также с целью повышения эффективности научно-исследовательской деятельности и ее вклада в повышение качества подготовки специалистов.

НАПРАВЛЕНИЕ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

Антонова Александра Денисовна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P41702
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: a79618072038@yandex.ru

Смолин Артем Александрович
Год рождения: 1977
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.философ.н.
e-mail: smolin@corp.ifmo.ru

УДК 000.00

ХУДОЖЕСТВЕННАЯ ВИЗУАЛИЗАЦИЯ В СРЕДЕ VR

Антонова А.Д.
Научный руководитель – к.философ.н. Смолин А.А.

Работа выполнена в рамках темы НИР «Методы визуализации художественных произведений в среде VR и программах 3D-моделирования».

В работе рассмотрены возможности создания графики в приложениях виртуальной реальности. Дана оценка существующих приложений, приведены характеристика и потенциал программ в области иллюстрирования и моделирования. С помощью гарнитуры HTC Vive проведена оценка интерфейса и управления некоторых приложений. Результатом стало выявление функциональных возможностей приложения Gravity Sketch, которое подходит для визуализации в художественной сфере.
Ключевые слова: виртуальная реальность (VR), визуализация, моделирование, гарнитура HTC Vive, иллюстрирование, приложение, медиаискусство.

С каждым годом виды искусства модернизируются и появляются новые названия, например, new media art (медиаискусство), иммерсивная живопись и использование 360 видео в искусстве. Технология виртуальной реальности (VR) позволяет художникам быстрее погрузиться в искусственную реальность, поэтому важно использовать удобные инструменты для воплощения идей. 3D-визуализация в искусстве также дает возможность увидеть любой художественный объект со всеми деталями, нюансами и под любым углом. Можно воссоздать абсолютно любой объект, даже не существующий в жизни, но сделать это так, что зритель поверит в его существование [1].

Исследование «Использование эталонных изображений и эскизов от руки для создания 3D-моделей в виртуальной реальности» рассматривает взаимодействие творческого потенциала человека и моделирования. Для точной визуализации необходимо долго разбираться в системе программного обеспечения. На это уходит время и часто желаемое невозможно воплотить из-за отсутствия навыков у художника.

Исследования показали, что взаимодействие со средой в VR быстрее погружает пользователя в интерфейс. Для создания эскизов в пределах VR требует от разработчиков создания системы контроля линий и примитивов, создаваемых пользователями, при этом оставляя стиль и правильные геометрические формы [2].

Потенциал VR как художественного инструмента дает возможность переосмыслить искусство. Сегодня уже существуют некоторые методы и инструменты визуализации для

демонстрации в VR. В основном этот контент создается в программах 3D-моделирования на компьютере. Визуализация в самой VR только начинает свое развитие. Тем не менее, уже появились редакторы для создания графики, моделирования и рисования. Сейчас многие деятели искусства экспериментируют и создают запоминающиеся виртуальные арт-проекты. Мэтт Шефер, Джама Джурабаев, Николас Бейкер создают в приложениях VR концепты, иллюстрации и модели промышленного дизайна, заменяя работу в программах на компьютере или соединяя эти процессы [3, 4].

Для наибольшего погружения используют гарнитуру (шлем) VR. На международном рынке производителей небольшое количество конкурирующих компаний, и все они находятся примерно на одном качественном и ценовом уровне. Поэтому изучение и тестирование допустимо проводить на любой гарнитуре. Использование HTC Vive для изучения приложений требует свободную зону 4,5×4,5 м и важные характеристики персонального компьютера: графический процессор: NVIDIA® GeForce® GTX 1060, AMD Radeon R9 290, аналогичная или более современная модель, и некоторые другие параметры, необходимые для хорошей и удобной работы со шлемом.

Используя метод анализа, выделены конкурентоспособные программы моделирования и визуализации в среде VR. В таблице представлены характеристики приложений, которые могут быть использованы в художественной сфере [5]. Проанализировав технические возможности и существующие примеры работ, особое внимание привлекает приложение компании Gravity Sketch и Tilt Brush.

Таблица. Обзор приложений

Характеристики	Приложения					
	Tilt Brush	Gravity Sketch	Unbound Alpha	Quill	Oculus Medium	Tvori
Экспорт и импорт файлов	Да (из Block)	Да	Да	Да	Да	Да
Совместная работа	–	Да	Да	–	–	–
Анимированные кисти	Да	Да	–	Да	Да	Нет
Анимация	Нет	Нет	–	Да	Нет	Да
Скульптинг	Нет	Да	Да	Нет	Да	Нет
Цена	7,00$	10,00$+(ежемесячная подписка)	30,79$	29,99$	29,75$	19,9$
Гарнитура	HTC Vive, Oculus	HTC Vive, Oculus	HTC Vive, Oculus	Oculus	Oculus	HTC Vive, Oculus

Tilt Brush – это создание трехмерной неподвижной графики с интуитивным интерфейсом и большим выбором кистей для рисования. Программа создана компанией Google с помощью художников, танцоров и других деятелей искусства.

Gravity Sketch – это интуитивно понятный инструмент для создания пользовательского интерфейса, разработанный специально для смешанной реальности. Создание 3D-моделей, концептуальные проекты интуитивно понятным способом с использованием технологий AR, VR и Touch. Позволяет быстро создавать свободную геометрию. Есть возможность импортировать модель напрямую в программу 3D.

Проведя эксперимент на интуитивно-понятный интерфейс, было определено, что управление и доступные функции действительно легко находятся самостоятельно, без дополнительного обучения. Тем не менее, разработчики добавили видеоконтент с обучением, которое можно использовать при возникновении проблем.

Перед испытуемым ставилась задача создать объемный пейзаж, на основе эскиза, показанного на рисунке. Испытуемый до эксперимента не был знаком с подобными программами, но опыт нахождения в виртуальном пространстве был. Инструменты редактирования и создания форм находятся на левом контролере, управление цветом, материалом и областью захвата объектов – на правом.

а б

Рисунок. Создание объекта: форма по эскизу (а); финальная форма (б)

Главная трудность, с которой сталкивается человек при использовании шлема – это усталость глаз и, как следствие, непродолжительное время использования. Для создания качественной картины необходимо делать перерывы в работе, а значит, скорость создания проектов в VR следует проверить.

VR идеально подходит для реализации творческих идей в области искусства. Следующим шагом к методике художественной визуализации является работа с выбранным программным обеспечением и увеличением количества испытуемых. Gravity Sketch на рынке имеет высокую оценку пользователей и оптимальную характеристику среди рассмотренных программ, и соревнуется только с Tilt Brush. Исследуя и изучая весь инструментарий программ, следует провести качественное исследование использования приложения художниками, цель которого выяснить, интерфейс какой из программы легче и удобнее в освоении. С помощью эксперимента, проводимого в гарнитуре VR, будут созданы одинаковые произведения живописи. Проведенный литературный обзор существующих исследований и экспериментов показал высокий потенциал развития VR в искусстве, а также роль новейших технологий в современной культуре.

Литература

1. Определение искусства виртуальной реальности [Электронный ресурс]. – Режим доступа: http://terme.ru/: http://terme.ru/termin/virtualnaja-realnost-v-iskusstve.html (дата обращения: 04.01.2019).
2. Jackson B., Keefe D.F. Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR // IEEE Transactions on Visualization and Computer Graphics. – 2016. – V. 22(4). – P. 1442–1451.
3. Работы Метт Шефера [Электронный ресурс]. – Режим доступа: http://www.mattschaeferdesign.com/vr-painting/ (дата обращения: 04.01.2019).
4. Работы Николаса Бейкера [Электронный ресурс]. – Режим доступа: https://nicholas-baker.com/ (дата обращения: 04.01.2019).
5. Приложения для рисования VR [Электронный ресурс]. – Режим доступа: https://artforlife.ru/:https://artforlife.ru/stati-po-dizajnu/7-luchshih-instrumentov-dlya-risovaniya-3d-modelirovaniya-i-skulptury-v-vr-osobennosti.html (дата обращения: 04.01.2019).

Бабарицкий Павел Александрович
Год рождения: 1994
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4262
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: redbear95@gmail.com

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.04

АНАЛИЗ И ИМПЛЕМЕНТАЦИЯ МЕТОДОВ ИЗВЛЕЧЕНИЯ И ОБРАБОТКИ ДАННЫХ В ВЕБЕ

Бабарицкий П.А., Государев И.Б.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

Работа выполнена в рамках темы НИР № 615892 «Исследование и разработка в области информационных технологий».

В работе изучался процесс развития обработки, извлечения и поиска информации. Были выделены методы извлечения и нахождения информации, их применение в веб, а также категории веб-извлечения и их особенности. Раскрыта связь между интеллектуальным анализом и синтаксическим анализом. На примере языка программирования раскрыт процесс web scraping и его возможности.
Ключевые слова: обработка информации, извлечение информации, поиск информации, парсинг, Web Mining, Web Scraping, интеллектуальный анализ данных, синтаксический анализ.

Главной задачей работы было, на основе существующих результатов, решающих проблему интеллектуального анализа данных:
– классифицировать Web Mining в зависимости от его особенностей;
– логически связать существующие понятия в исследуемой области;
– описать процесс WebScraping и инструменты языка PHP, использующиеся для его реализации.

Так как деятельность в области интеллектуального анализа не переставала развиваться по мере развития сети, то это позволило на основе рассмотрения источников по различным выделенным категориям обработчиков информации обобщить их в табл. 1, отразив в ней виды данных: с какой информацией осуществляется работа, как они представлены, какие методы используются и к каким категориям относятся создаваемые приложения [1, 2].

В процессе раскрытия теоретических основ извлечения информации из веб-ресурсов и процесса их развития, была обнаружена проблема, связанная с многообразием используемых понятий. В связи с опорой на научные работы [3, 4] и результаты предыдущего исследования [5], была создана схема, изображенная на рис. 1. Из нее становится понятно, чем отличаются и как связаны различные понятия, имеющие отношение к теме обработки и извлечения данных. После чего исследование перешло от рассмотрения обработки информации в общем

к частному, сосредоточилось на исследовании процесса Web Scraping, также отраженного на полученной схеме. Для этого были разработаны простейшие приложения на языке PHP, решающие различные поставленные проблемы.

Таблица 1. Категории извлечения веб-данных

	Web Mining			
	Web Content Mining		Web Structure Mining	Web Usage Mining
	Информационный поиск	Поиск в базе данных		
Вид данных	– не структурированные; – полуструктурированные	– полуструктурированные; – веб-сайт как база данных	– структура ссылок	– интерактивное
Данные	– текстовые документы; – гипертекстовые документы	– гипертекстовые документы	– структура ссылок	– логи серверов; – логи браузеров
Представления	– набор слов, N-грамма; – термины, фразы; – концепция или онтология; – реляционный	– граф с разметкой ребер (OEM); – реляционная модель данных	– граф	– реляционная таблица; – граф
Методы	– мера TF-IDF и вариации; – машинное обучение; – статистические (включая нейролингвистическое программирование (НЛП))	– собственные (патентованные) алгоритмы; – индуктивное логическое программирование (ИЛП); – (модифицированные) ассоциативные правила обучения	– собственные (патентованные) алгоритмы	– машинное обучение; – статистические; – (модифицированные) ассоциативные правила обучения
Задачи приложения	– категоризация; – кластеризация; – поиск правил извлечения; – поиск шаблонов в тексте; – адаптация систем к конкретным потребностям пользователя (человеко-компьютерное взаимодействие)	– поиск часто встречающихся подструктур; – исследование схемы веб-сайта	– категоризация; – кластеризация	– разработка сайта, адаптация и управление; – маркетинг; – адаптация систем к конкретным потребностям пользователя (человеко-компьютерное взаимодействие)

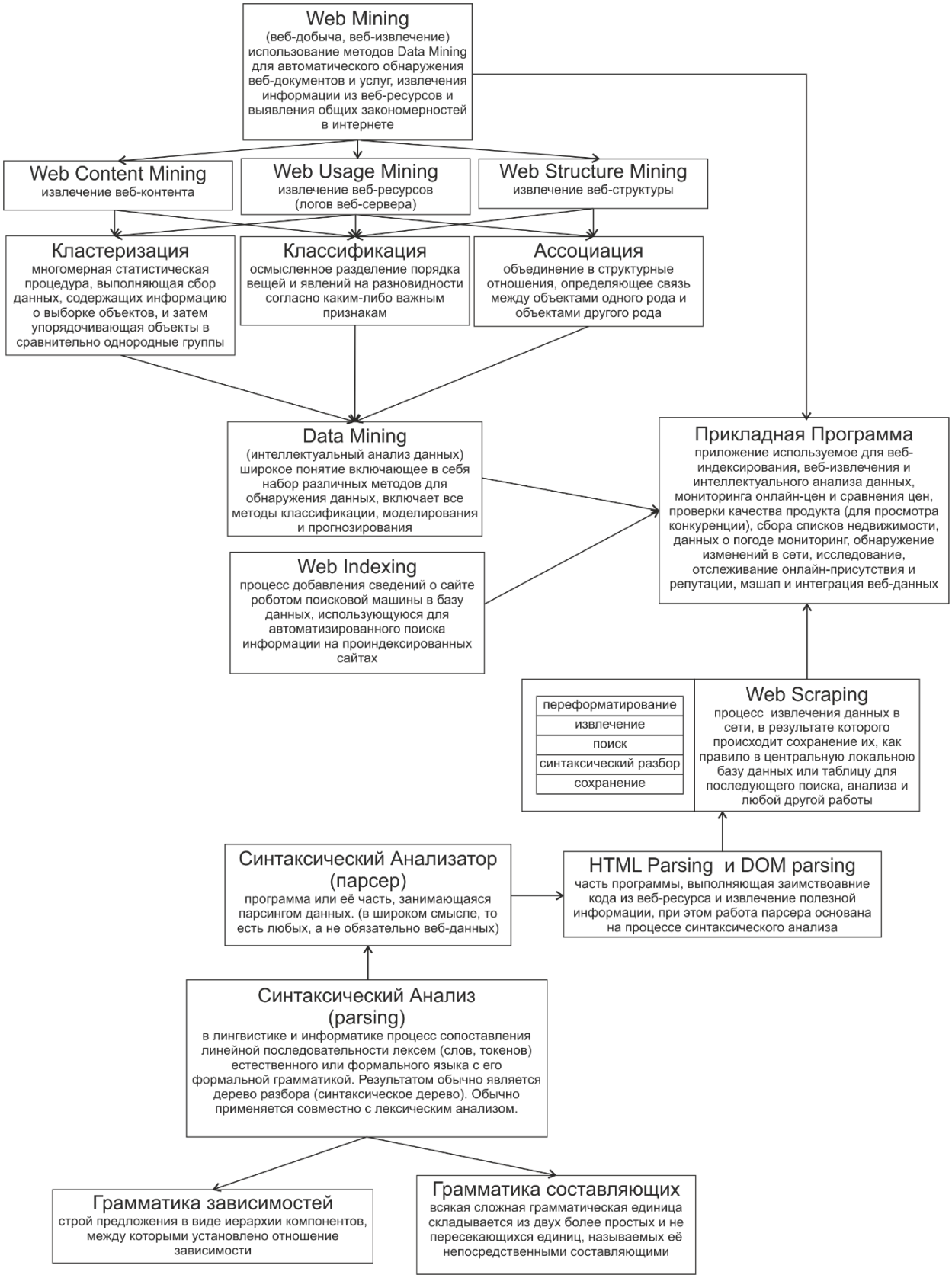


Рис. 1. Схема взаимосвязей между синтаксическим анализом, веб-извлечением данных и интеллектуальным анализом данных

В процессе разработки и рассмотрения приложений, занимающихся процессом извлечения и обработки данных, были закреплены задачи, которые решает язык PHP и его инструменты, в табл. 2 [6, 7].

Таблица 2. Возможности языка PHP

Задачи, решаемые в PHP	Инструменты
Извлечение данных по запросу	1. функция file_get_contents (); 2. библиотека cURL
Обработка полученной информации	1. пользовательские функции; 3. внутренняя библиотека PHP XPath DOM; 4. внешние библиотеки: PHP Simple HTML DOMParser, Zend_Dom_Query, phpQuery, DOMQuery и другие; 5. функции для работы со строками и декодирования данных
Сохранение результатов	1. функции работы с файлами; 2. интерфейс PDO для работы с базами данных; 3. переменные PHP
Обработка нескольких страниц	1. извлечение данных с использованием циклов; 2. функции библиотеки cURL для работы с мультисессией
Повторное использование типизированных решений очистки данных	1. объектно-ориентированное программирование

Изучение возможностей исследуемого языка позволило получить представление по осуществлению работы с данными в Интернете, в итоге была составлена схема, демонстрирующая такой исследуемый процесс, как Web Scraping в общих чертах (рис. 2).

Данная схема объясняет, в чем заключается процесс Web Scraping. В качестве источника данных может служить веб-сайт или несколько сайтов. Процесс работы с данными реализуется с помощью языка программирования, а сам процесс Web Scraping заключается в обращении к источнику данных для извлечения информации, которая затем обрабатывается и в нужном формате передается в хранилище данных.

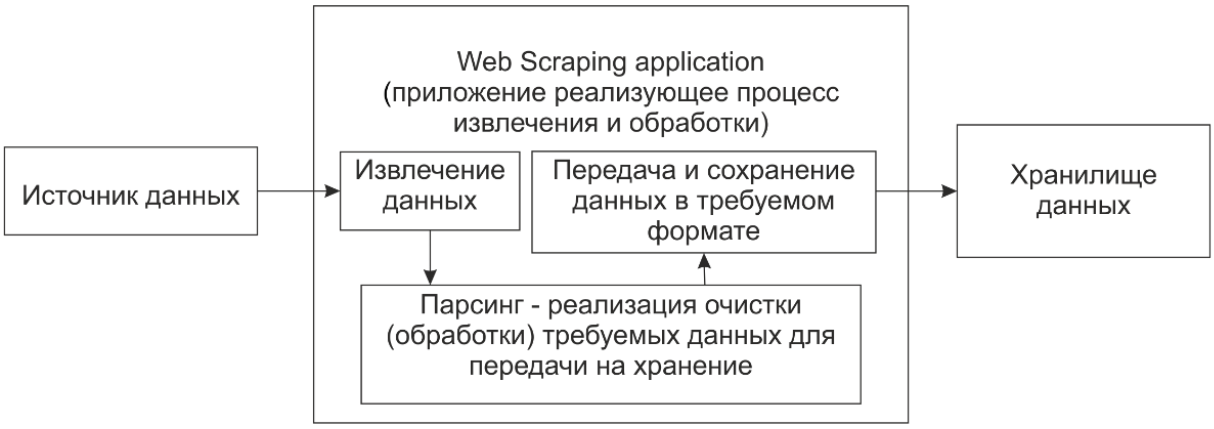


Рис. 2. Web Scraping

В результате проведенного исследования были раскрыты как теоретические, так и практические аспекты изучаемой проблемы извлечения и обработки информации:

– выделены парадигмы машинного обучения;
– рассмотрены существующие технологии по работе с данными;
– классифицирован Web Mining;
– раскрыты используемые теоретические понятия;
– отражена суть и реализация такого процесса, как Web Scraping с использованием языка PHP;
– выделены инструменты, которыми обладает язык PHP для работы с информацией.

Литература

1. Voorhees E., Harman D. Information Processing & Management // Overview of the Sixth Text REtrieval Conference (TREC-6). – 1998. – P. 1–24.
2. Trybula W.J. Text mining // Annual Review of Information Science and Technology. – 1999. – V. 34. – P. 385–419.
3. Etzioni O. World-Wide Web: Quagmire or gold mine // Communications of the ACM. – 1996. – V. 39(11). – P. 65–68.
4. Cooley R., Mobasher B., Srivastava J. Web mining: information and pattern discovery on the World Wide Web // Proceedings of the 9th ZEEE International Conference on Tools with Artificial Intelligence. – 1997. – P. 558–567.
5. Бабарицкий П.А. Анализ и имплементация методов парсинга веб-ресурсов. анализ зарубежных и отечественных источников по теме исследования // Альманах научных работ молодых ученых Университета ИТМО. – 2018. – Т. 7. – С. 51–53.
6. Трепачёв Д.П. Учебники, задачники, справочники по web языкам [Электронный ресурс]. – Режим доступа: http://theory.phphtml.net (дата обращения: 18.12.2018).
7. Алексеев Д.М., Кутняк Н.А. Применение возможностей языка программирования PYTHON для парсинга web-страниц, требующих авторизации // Приоритетные направления развития науки, техники и технологий международная научно-практическая конференция. – 2016. – С. 62–64.

Бабахин Антон Валерьевич
Год рождения: 1994
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4263
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: anton-098@yandex.ru

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.5

СРАВНИТЕЛЬНЫЙ АНАЛИЗ CSS-МЕТОДОЛОГИЙ

Бабахин А.В., Государев И.Б.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

В работе определены основные сложности, с которыми сталкиваются разработчики веб-интерфейсов, а также описан основной подход к их решению. Выявлены самые популярные CSS-методологии, выполнен их обзор. Определены критерии сравнения методологий и выполнен сравнительный анализ. Даны рекомендации по использованию методологий верстки в проектах.
Ключевые слова: CSS-методологии, пользовательский интерфейс, BEM, Atomic CSS, SMACSS, OOCSS.

В начале развития сети Интернет веб-страницы представляли собой наборы текстов, картинок и гиперссылок. В процессе эволюции структуры сайтов становились сложнее, оформление включало новые элементы, увеличивалась активность взаимодействия страниц с посетителем. Соответственно, сложность разработки пользовательских веб-интерфейсов также возросла, код CSS-файлов насчитывает тысячи и десятки тысяч строк, над интерфейсом работают команды разработчиков.

При разработке пользовательского интерфейса приходится сталкиваться со следующими проблемами: сложность подключения новых сотрудников к проекту, трудность анализа исходного кода и поиска ошибок в нем [1].

Для решения данных проблем применяют CSS-методологии – наборы правил по организации кода. Это универсальный подход для разработки современных веб-интерфейсов.

На данный момент не существует каких-либо научных исследований, посвященных сравнительному анализу CSS-методологий. Решение о применении определенной методологии в проекте зачастую основано на прошлом опыте разработчика или его предпочтениях. Таким образом, подтверждается актуальность данного исследования, результаты которого могут способствовать аргументированному выбору методологий верстки в реальных проектах.

Самыми популярными CSS-методологиями на сегодняшний день являются:

- BEM;
- Atomic CSS;
- OOCSS;

– SMACSS.

Методология BEM была представлена российской компанией Яндекс в 2009 году, и сегодня она является самой популярной [2]. Главными элементами BEM являются: блок, элемент и модификатор. Ключевая особенность методологии – поддержка модульности.

Блок – независимый, многократно используемый компонент страницы. Имя блока включается в названия элементов и модификаторов, тем самым создавая пространство имен. Благодаря этому подходу вероятность совершения ошибки, связанной с присвоением двум элементам одинаковых классов, существенно уменьшается. Таким образом, блок можно многократно использовать не только на одной странице, но и в разных проектах.

Составной частью блока является элемент. Название элемента отделяется от имени блока двумя знаками нижнего подчеркивания. Не допускается использование элементов вне блока, с которым они связаны.

Модификаторы применяются для задания блокам или элементам особого оформления. Ключи модификаторов и их значения отделяются при помощи одного знака нижнего подчеркивания.

Методология Atomic CSS была предложена компанией Yahoo в 2015 году. Особенностью данной методологии является перенос описания стилей из каскадных таблиц стилей в код HTML. Общее оформление каждого элемента страницы является результатом наложения небольших изменений [3].

Благодаря данному подходу существенно уменьшается объем CSS-кода. Обратной стороной применения методологии является увеличение кода HTML, а также сложность его анализа.

В отличие от написания inline-стилей, классы Atomic CSS поддерживают псевдо-классы и медиазапросы. Для извлечения таких классов из кода HTML применяется специальный инструмент – Atomizer.

Методология OOCSS появилась в 2013 году. В ее основе лежит объектно-ориентированный подход, что свидетельствует о наличии независимых компонентов. Данные компоненты могут быть использованы многократно, что позитивно сказывается на объеме CSS-кода [4].

Правила данной методологии запрещают делать код стилей одних элементов сильно зависимым от других элементов. Благодаря этому упрощается внесение изменений и рефакторинг.

CSS-методология SMACSS была создана в 2011 году. В основе ее модульной и масштабируемой архитектуры лежит категоризация стилей [5].

Определено пять категорий:

– Base;
– Layout;
– Module;
– State;
– Theme.

В стилях базовой категории используются только селекторы тегов, селекторы псевдо-элементов и псевдо-классов, а также селекторы потомков.

Layout-стили определяют оформление структурных блоков веб-страницы. Для этой цели применяются селекторы id.

Стили модулей предназначены для оформления элементов, встречающихся на странице один или более раз. Поэтому в данной категории используются селекторы классов. Для изменения варианта отображения модуля используются стили состояния.

Благодаря стилям темы имеется возможность существенного изменения отображения страниц сайта. Названия классов данной категории должны начинаться с префикса theme-.

Таким образом, после проведения обзора CSS-методологий OOCSS, BEM, Atomic CSS и SMACSS, были определены следующие критерии их сравнения: объем CSS-кода, объем HTML-кода, простота создания мобильной версии, простота создания адаптивной верстки, возможность оптимизации под браузеры, повторное использование кода, количество документации, простота сборки проекта, удобство применения специальных инструментов разработки, количество плагинов и SDK.

Для оценки каждого критерия использовалась шкала от 1 до 10.

Результаты сравнительного анализа представлены в таблице.

Таблица. Сравнение CSS-методологий

Критерий сравнения	BEM	Atomic CSS	OOCSS	SMACSS
Объем CSS-кода	8	10	9	8
Объем HTML-кода	7	6	8	8
Простота создания мобильной версии	10	6	8	8
Простота создания адаптивной верстки	8	9	8	8
Возможность оптимизации под браузеры	9	6	8	9
Повторное использование кода	10	7	9	8
Количество документации	10	8	8	9
Простота сборки проекта	7	8	9	9
Удобство применения специальных инструментов разработки	10	7	8	8
Количество плагинов и SDK	10	9	9	9
Итого	89	76	84	84

Таким образом, в результате проведенного сравнения можно сделать вывод, что наибольшим количеством плюсов обладает методология BEM, набравшая 89 баллов. Благодаря тому, что она является самой популярной, на сегодняшний день существует множество инструментов разработки, SDK, плагинов и документации.

Одинаковый результат получили методологии OOCSS и SMACSS – 84 балла. Как видно из сравнения, по всем критериям у данных методологий получено не менее 8 баллов, что свидетельствует об универсальности данных подходов для разработки интерфейсов.

Наименьшее количество баллов имеет методология Atomic CSS – 76 баллов. Таким образом, можно сделать вывод, что применять данный подход целесообразно только в крупных проектах.

В качестве направления дальнейших исследований предполагается разработать новую методологию верстки, обладающую конкурентными преимуществами, по сравнению с рассмотренными ранее.

Литература

1. Решение распространенных проблем веб-разработки с помощью БЭМ [Электронный ресурс]. – Режим доступа: https://ru.bem.info/methodology/solved-problems/ (дата обращения: 04.01.2019).
2. Способы организации CSS-кода [Электронный ресурс]. – Режим доступа: https://habrahabr.ru/post/256109/ (дата обращения: 04.01.2019).
3. Thinking in Atomic | Atomic CSS [Электронный ресурс]. – Режим доступа: https://acss.io/thinking-in-atomic.html (дата обращения: 04.01.2019).
4. Object Oriented CSS [Электронный ресурс]. – Режим доступа: https://github.com/stubbornella/oocss/wiki (дата обращения: 04.01.2019).
5. Snook J. Scalable and Modular Architecture for CSS: A Flexible Guide to Developing Sites Small and Large. – Snook Ca Web Development, 2012. – 4 p.

Баев Андрей Александрович
Год рождения: 1996
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4260
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: ghost9610@yandex.ru

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.02

МОДЕЛИРОВАНИЕ СИСТЕМЫ РЕЙТИНГА В МНОГОПОЛЬЗОВАТЕЛЬСКИХ ИГРАХ НА ВЕБ-ПЛАТФОРМЕ

Баев А.А., Государев И.Б.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

В работе представлен анализ проблемы влияния на рейтинг игрока факторов, которые не зависят от игрока. Рассмотрены примеры факторов, которые влияют на результат игры. Приведен авторский принцип построения системы рейтинга через отказ от результата игры.
Ключевые слова: компьютерные игры, разработка игр, рейтинговые системы, рейтинг, проблемы проектирования игр.

Хорошо спроектированная рейтинговая система позволяет улучшить аспект балансировки игр и разрешить проблемы, возникающие при этой балансировке. Jesse Schell [1] и Tynan Sylvester [2], приводят примеры, из которых можно заметить, что игры с плохим балансом чаще обречены на неудачные продажи и порицание со стороны сообщества игроков.

На сегодняшний день рейтинг игрока в компьютерных играх часто зависит от результата игры – победы. Результат игры, в свою очередь, зависит от многих других факторов, как игровых, так и не игровых. Среди этих факторов могут оказаться такие факторы, которые не зависят напрямую от игрока, чей рейтинг система будет изменять в результате игровой партии. Другими словами, рейтинг игрока не полностью зависит от игрока, качества его умений. Из-за этого возникает проблема неточной оценки умений игрока, его рейтинга.

Команда League of Legends выделяет [3, С. 1] следующие факторы, от которых зависит рейтинг игрока:

- выбрана новая роль в игре;
- выбрана новая тактика;
- порой игроки просто не в форме.

Последний фактор достаточно важен. Сервис porofessor.gg [4], который предоставляет анализ каждого матча, приводит следующие факторы, влияющие на «форму игрока»:

- плохое настроение;
- подавленный;
- любимый персонаж игры заблокирован;
- выбрана новая роль в команде;

– игрок плох против выбранного соперником персонажа;
– первая игра этого дня;
– серия побед;
– серия поражений.

Следует отметить, что сервис porofessor.gg анализирует около 2 700 000 матчей в день, поэтому можно доверять приведенным им факторам.

Также в командных играх часто возникает проблема умышленного саботирования победы своей команды одним или несколькими игроками этой команды. Для такого поведения в некоторых играх существуют такие термины как «умышленное раскармливание» и «неспортивное поведение», а также методы жалоб на таких игроков, однако, факт умышленного саботажа не всегда легко доказать.

Стоит отметить, что все эти факторы так или иначе влияют на результат игры, однако, они не зависят напрямую от действий игрока.

Одним из решений этой проблемы может быть отказ от влияния фактора победы, т.е. рейтинг игрока не будет зависеть напрямую от результата игры, а лишь от тех факторов, которые зависят от действий самого игрока. В таком случае рейтинг игрока будет показывать более точную оценку его умений.

При использовании этого принципа уравнение расчета изменения рейтинга игрока примет другой вид. Оно будет включать в себя лишь те переменные и коэффициенты, которые будут отвечать непосредственно за действие игрока.

При проектировании системы рейтинга, учитывая этот принцип, как и при проектировании системы в целом – отдельные факторы, оказывающие влияние на изменения рейтинга, и коэффициенты их влияния полностью выбираются разработчиком.

Можно привести пример, какие факторы могут быть использованы при учете изменения рейтинга:

– точность стрельбы в играх, связанных со стрельбой;
– количество действий с помощью компьютерной мыши в играх-стратегиях;
– количество полученного золота в MOBA-играх (Multiplayer Online Battle Arena игры);
– фактор первой игровой сессии после долгого перерыва (игра без предварительной разминки, например, снижает коэффициенты влияния от других факторов).

И многие другие факторы в зависимости от условий и правил игры, состава игровой логики и пожеланий разработчика.

Таким образом, можно решить проблему влияния факторов на рейтинг, которые не зависят от игрока.

Литература

1. Schell J. The Art of Game Design: A Book of Lenses [Электронный ресурс]. – Режим доступа: http://sg4adults.eu/files/art-game-design.pdf (дата обращения: 10.01.2019).
2. Sylvester T. Designing Games: A Guide to Engineering Experiences, 2013 [Электронный ресурс]. – Режим доступа: http://www.reedbushey.com/32Designing%20Games.pdf (дата обращения: 10.01.2019).
3. Riot Games Team, Matchmaking Guide [Электронный ресурс]. – Режим доступа: https://support.riotgames.com/hc/en-us/articles/201752954-Matchmaking-Guide (дата обращения: 10.01.2019).
4. Porofessor, League of Legends live game search and real-time player statistics [Электронный ресурс]. – Режим доступа: https://porofessor.gg/ (дата обращения: 10.01.2019).

Балаев Ариф Камилевич
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P4262
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: balaev.arif@yandex.ru

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.451.4

МЕТОДОЛОГИИ РЕФАКТОРИНГА УНАСЛЕДОВАННОГО КОДА. ПОДГОТОВИТЕЛЬНАЯ ФАЗА РЕФАКТОРИНГА

Балаев А.К., Государев И.Б.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

Работа выполнена в рамках темы НИР № 615892 «Исследование и разработка в области информационных технологий».

В работе рассмотрена проблема некорректной реализации рефакторинга, в процессе которого система начинает вести себя иначе. Определены базовые пункты, необходимые до начала рефакторинга, для осуществления безрискового изменения кода.
Ключевые слова: унаследованный код, рефакторинг, подготовительная фаза рефакторинга, архитектура проекта.

Зачастую этап разработки сталкивается с ситуацией невозможного продолжения расширения функционала по причинам некорректной архитектуры, большого количества унаследованного трудночитаемого кода и специфики реализации той или иной бизнес-логики.

При возникновении данной проблемы рефакторинг неизбежен. Он позволяет сделать внутренность системы понятнее и более структурированной. Критерием успешности проведения рефакторинга является полная сохранность поведения информационной системы с внешней средой. Однако перед тем как менять структуру кода, следует покрыть все риски его провальной реализации.

В сфере разработки программного обеспечения нередким случаем является возникновение таких ситуаций после рефакторинга как появление ранее не существующих нарушений в старой бизнес-логике и невозможность выпустить продукт из-за его неработоспособности, созданной рефакторингом [1]. Исходя из этого, проблема ошибочной имплементации рефакторинга будет актуальна до тех пор, пока все, кто принимается за работу с унаследованным кодом, не будут покрывать риски и осуществлять подготовительную фазу до ее старта.

Целью работы являлось выявление минимальных необходимых элементов, которые позволят осуществить рефакторинг наиболее безопасным образом, как для бизнеса, так и для разработки.

Для достижения данной цели были поставлены следующие задачи:

– выявить фундаментальные критерии безопасного проведения рефакторинга;
– определить инструменты, позволяющие реализовать данные критерии.

Как показывает практика при рефакторинге системы необходимо полагаться на три фундаментальных принципа:

1. система должна продолжать вести себя полностью идентично;
2. всегда должен быть план Б, для отката системы в полное работоспособное состояние;
3. рефакторинг производить небольшими итерациями, позволяющими менять структуру кода с полной уверенностью о сохранности системы.

Для того чтобы быть уверенным в сохранности поведения системы, необходим инструмент, проверяющий, ведет ли себя система идентично после каждой итерации изменений. Идеальным вариантом в таком случае являются автоматически запускаемые тесты [2]. Данные тесты должны быть написаны до старта рефакторинга и должны покрывать всю основную бизнес-логику приложения. В случае успешного прохождения тестов до и после изменения кода при грамотном написании тестов риски неисправности системы будут гораздо снижены.

При написании тестов до рефакторинга многие разработчики допускают ошибку и пишут тесты на внутренние методы системы. Тем самым, поддаваясь реструктуризации, помимо самой логики приложения необходимо и реструктурировать методы тестов, что является полностью некорректным их использованием [3]. Необходимо писать тесты на методы, которые гарантированно не изменятся в своей сигнатуре, списков параметров в виде выходных данных. Единственными из всех видов методов, которые подходят под описанный критерий, являются публичные методы приложения, использующиеся для коммуникации со сторонними системами [4].

Во избежание невозможности предоставить рабочую версию приложения в период рефакторинга, необходимо всегда иметь возможность откатиться на несколько шагов назад, приэтом не теряя недоделанную работу. Современную разработку информационных систем уже сложно представить без систем контроль версии, но встречаются случаи ее подключения после того, как работа была начата. Система контроля версии позволяет не только упростить совместную разработку проекта нескольким разработчикам, но и полностью покрыть риски неисправной версии проекта на соответствующий период времени.

Последним пунктом, необходимым до начала изменения кода, является регулярный запуск проверяющих тестов на любое изменение в системе контроля версии [5]. Данный кейс позволяет в короткие сроки убедиться, что система ведет себя идентичным образом. В обратном случае разработчик сразу же сможет исправить образовавшуюся ошибку или вернуться на версию раньше.

В результате данной работе были определены и сформулированы фундаментальные пункты, необходимые до старта рефакторинга и покрывающие основную часть рисков изменения поведения системы во время изменения внутренней структуры кода.

Литература

1. Ксензов М.В. Рефакторинг архитектуры программного обеспечения [Электронный ресурс]. – Режим доступа: http://www.ispras.ru/ru/proceedings/docs/2004/8/1/isp_2004_8_1_211.pdf (дата обращения: 04.01.2019).
2. Назаров С.В. Архитектура и проектирование программных систем. – М.: ИНФРА-М, 2013. – 351 с.
3. Фаулер М. Рефакторинг: улучшение существующего кода / Пер. с англ. – СПб.: Символ Плюс, 2003. – 432 с.
4. Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста. – СПб.: Питер, 2013. – 464 с.
5. Физерс М.К. Эффективная работа с унаследованным кодом / Пер. с англ. – М.: Вильямс, 2009. – 400 с.

Рущенко Нина Геннадиевна
Год рождения: 1972
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: rushchenko@corp.ifmo.ru

Барышева Кристина Владимировна
Год рождения: 1990
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41721
Направление подготовки: 09.04.02 – Дизайн человеко-компьютерных систем
e-mail: unionfourman@gmail.com

УДК 004.51

АНАЛИЗ ПРИМЕНЕНИЯ ПЕРЕИСПОЛЬЗУЕМЫХ ЭЛЕМЕНТОВ В СОВРЕМЕННЫХ СИСТЕМАХ ДИСПЕТЧЕРСКОГО КОНТРОЛЯ И СБОРА ДАННЫХ

Рущенко Н.Г., Барышева К.В.
Научный руководитель – к.т.н., доцент Рущенко Н.Г.

В работе рассматривались различные программные пакеты, предназначенные для разработки систем мониторинга и (или) управления с целью сбора, обработки, отображения и архивирования информации об объекте. В ходе исследования был проведен сравнительный анализ применяемых способов повторного использования элементов (по экземпляру, параметризацией, путем адаптации) в существующих SCADA.
Ключевые слова: SCADA, система диспетчеризации, редактор, библиотека элементов, повторно используемые элементы.

Системы диспетчерского контроля и сбора данных представляют собой критически важную, динамично развивающуюся инфраструктуру, составляющую глобальной экономики, ввиду автоматизации всех процессов производства и потребления. Конечная цель любой SCADA-системы – предоставление человеко-компьютерного интерфейса (ЧКИ) – позволяющего производить оперативный мониторинг и управление автоматизированной системой. При проектировании визуализации данных систем разработчики должны решить ряд задач: усложнение работы с оборудованием, связанное с модернизацией уже эксплуатируемой системы автоматики, проблемами эргономичности интерфейса, так как развитие технологий позволяет подключать к системам все большее количество единиц оборудования, а также включать в проект множество параметров отладки, что должно учитываться в конечном человеко-компьютерном взаимодействии (Human-Computer Interaction, HCI).

В результате анализа и структурирования обобщенно можно описать три вида способа повторного использования элементов [1]:

– повторное использование по экземпляру (пользователь самостоятельно определяет, какой блок и как он хочет повторно использовать);

– повторное использование параметризацией (у изначальных блоков лишь обозначены параметры, но не заданы их характеристики (количественные, качественные), пользователь сам задает их с учетом текущего проекта);
– повторное использование путем адаптации (берутся элементы уже из готовой библиотеки и редактируются пользователем).

Цель работы – проанализировать программные пакеты на применение повторного использования объектов при проектировании SCADA.

Для достижения этой цели необходимо было провести обзор и сравнительный анализ (таблица) программных пакетов, позволяющие разрабатывать системы ЧКИ [2]. Были рассмотрены следующие программные пакеты:

1. Simatic WinCC (Siemens, Германия);
2. MasterScada (ИнСАТ, Россия);
3. Intouch, (Wonderware, США);
4. Simple-Scada (Симпл-Скада, Россия);
5. EasyBuilder Pro (Weintek, Тайвань).

Таблица. Сравнительный анализ

Параметры	Simatic WinCC	MasterScada	Intouch	Simple-Scada	EasyBuilder Pro
По экземпляру	+	–+	–+	–	–
Визуальные компоненты	+	–	+	–	–
Функциональные компоненты	+	–	+	–	–
Пользовательские компоненты	+	+	–	–	–
Адаптация	–+	–+	–+	–+	–+
Визуальные компоненты	+	+	+	+	+
Функциональные компоненты	+	+	+	+	+
Пользовательские компоненты	–	–	–	–	–
Параметризация	–+	–+	–+	–	–+
Визуальные компоненты	–	–	–	–	–
Функциональные компоненты	+	+	–	–	+
Пользовательские компоненты	+	–	+	–	–

Повторное использования путем адаптации – наиболее часто используемый принцип, на этом принципе строится библиотека стандартных элементов (как визуальных, так и функциональных), но при этом не все редакторы обладают возможность создания пользовательской библиотеки.

Повторное использование путем параметризации: отвечает, как правило, за стандартные функции SCADA-проекта, такие как архивирование, работа с тревогами, аудит. Иначе говоря, возможность кастомизации визуальных компонентов практически везде отсутствует, но при этом предусмотрено повторное использование функциональных блоков, в некоторых редакторах также представлен редактор скриптов, позволяющий создавать пользовательские элементы (отчеты, уведомления и т.д.).

Повторное использование по экземпляру наиболее редко встречается в редакторах. Так как он наиболее трудозатратный, с точки зрения разработчика такого редактора, но при этом он позволяет максимально быстро вносить любые изменения в существующий проект, либо масштабировать его, как показывает анализ – этот метод применяется лишь в одном рассматриваемом продукте, разработчиком которого является один из лидеров промышленной автоматики в мире.

Литература

1. Sharmin M., Bailey B.P., Coats C., Hamilton K. Reuse of Design Objects in CAD Frameworks // Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design. – 1994. – P. 754–761.
2. Елизаров И.А., Третьяков А.А., Пчелинцев А.Н. и др. Интегрированные системы проектирования и управления: SCADA-системы: учебное пособие. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2015. – 160 с.

Безбах Юлия Игоревна
Год рождения: 1992
Университет ИТМО, факультет программной инженерии
и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: jul-ka5@mail.ru

Смолин Артем Александрович
Год рождения: 1977
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.философ.н.
e-mail: smolin@corp.ifmo.ru

УДК 004.94

ОРГАНИЗАЦИЯ ДОСТУПНОЙ СРЕДЫ ДЛЯ СЛАБОВИДЯЩИХ СРЕДСТВАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Безбах Ю.И.
Научный руководитель – к.философ.н. Смолин А.А.

В работе рассмотрены существующие технологические подходы к решению проблем слабого зрения и проблем людей с ограниченными возможностями по зрению. Исходя из существующих финансовых и технических предпосылок, решено было рассмотреть вариант разработки концепции модели очков для юридически слепых людей на основе технологии дополненной реальности.
Ключевые слова: дополненная реальность, локализация объекта в видеопотоке, умные очки, системы неполных данных, носимые устройства, доступная среда.

Технологии дополненной реальности позволяют реализовать представление дополнительной информации о реальных объектах окружающего пространства. Технологии широко применяются как в игровой индустрии, так и для тяжелого производства. Для решения определенной задачи создается приложение [1]. Пользователи систем дополненной реальности воспринимают как единое целое реальные объекты и синтезируемую системой визуальную, аудиальную, иногда даже тактильную информацию. Современные разработки находят применение в области новых медицинских технологий и научно-практической медицине. Сегодня существует большой выбор различных средств и протезов для людей со слабым зрением. Все они коммерческие и стоят больших денег, которых у людей с ограниченными возможностями в большинстве своем нет. Более того, многие люди со сниженным зрением по причине старения не считают себя инвалидами. Начиная примерно с 40 лет, у людей ухудшается зрение, в том числе снижается способность фокусироваться на близком тексте, уменьшается восприятие цвета, уменьшается контрастная чувствительность. Эти ухудшения не вызваны болезнью, проблемы, связанные с резкостью, фокусировкой подлежит коррекции с помощью очков или большего количества освещения, но есть проблемы, которые невозможно исправить привычными методами, например, потеря контрастности.

В 1972 году Всемирная организация здравоохранения (ВОЗ) приняла следующее определение: человек юридически считается слепым, если острота центрального зрения в условиях максимальной коррекции не превышает 3/60. При таком зрении человек в условиях дневного освещения с максимальной коррекцией оптики неспособен сосчитать пальцы с расстояния в 3 м. «Низкое зрение» относится к нарушениям зрения, отличным от слепоты. Слабое

зрение включает нарушения, которые не исправляются обычными очками, контактными линзами, лекарствами или операциями. В этих условиях слабое зрение часто определяется как нарушение зрения, которое препятствует способности человека выполнять повседневные действия. В информационном бюллетене по нарушениям зрения и слепоте [WHO-VI] ВОЗ подсчитала, что в мире насчитывается 246 млн человек со слабым зрением и 39 млн слепых, что указывает на то, что 86% людей с нарушениями зрения имеет слабое зрение. Эти данные свидетельствуют о том, что проблема является актуальной. Существуют фонды, занимающиеся вопросами людей с ограниченными возможностями по зрению (рисунок).

Рисунок. Количество людей в мире с ограничениями по зрению

Государственная программа Российской Федерации поддерживает программу «доступная среда» с 2011 года. К сожалению, в условиях большого государства сложно создать всеобъемлющую систему навигации для слабовидящих, помимо использования звуковых сигналов на светофорах. Поэтому удобнее использовать индивидуальные устройства. К тому же слабовидящих намного меньше, чем пешеходных переходов, и снабжение пешеходов индивидуальными носимыми устройствами экономически эффективно. Согласно специалисту светофорного хозяйства города Санкт-Петербурга для постройки светофорных объектов на перекрестке или площади требуется от 2,5 до 10 млн руб. Стоимость же одного проектируемого прибора не должна превышать 35 тыс. руб.

Технически существует не так много средств, которые могут существенно улучшить положение человека с ограниченными возможностями по зрению. Условно их можно разделить на три основные группы:

1. камеры с набором дополнительных функций;
2. проекторы, передающие информацию непосредственно на сетчатку глаза;
3. бионические протезы.

Последние представляют раздел кибернетики. В этой области ведутся широкомасштабные разработки на протяжении последних двадцати лет, но эффективно функционирующий бионический глаз ученым разработать не удалось. Существует несколько видов имплантатов. Цель создания электронного глаза – помощь слабовидящим с проблемами сетчатки или зрительного нерва. Имплантируемые вместо поврежденной сетчатки устройства должны заменить миллионы клеток фоторецепторов глаза [2].

Другие устройства, работающие по принципу проектора, существуют не так давно, и представляют собой разработки 2016–2018 гг. Устройства оснащены крошечной камерой и мини-лазером, который захватывает изображение – будь то видео, страница в интернете или книге – и проецирует его на сетчатку, как на экран. Сетчатка преобразует эти сигналы, передает их по зрительному нерву в мозг, а он трансформирует их в изображения.

Другой подход – использование камеры, датчиков положения, движения, магнитных датчиков и прочих вспомогательных устройств для анализа окружения, и транслирования информации с помощью различных каналов. Так носимое устройство искусственного зрения, предназначенное для помощи слепым, слабовидящим или имеющим проблемы с чтением при помощи интеллектуальной камеры мгновенно считывает печатный и цифровой текст вслух, распознает лица, продукты и денежные записи в реальном времени.

В работе рассмотрена модулируемая система, которая с помощью технологии дополненной реальности может обнаруживать и выделять близлежащие объекты. Система выполняет данные операции за счет повышения контрастности изображения, переводя его в градации серого, и особенно контрастно выделяет контур. Помимо этого, система обозначает значимые объекты и предметы при помощи системы маркеров.

Отделы мозга, которые отвечают за работу зрения (подкорковый зрительный, теменная доля, затылочная доля), продолжают функционировать даже тогда, когда оно практически отсутствует [2]. Предполагается создание очков, оснащенных встроенной видеокамерой и датчиками для возможности детальной конфигурации предметов. Затем съемки оцифровываются, информация обрабатывается [3], программное обеспечение сортирует ее таким образом, что для того, кто воспользуется такими очками, дополненная реальность воспринимается как действительность. Изображение упрощается, и за счет упрощения и удаления повседневного «шума» человеку с ограниченными возможностями по зрению поступает четкое, контрастное изображение, содержащее только первостепенную информацию. Так, например, быстродвижущийся в направлении смотрящего объект будет многократно визуально усилен. Также проект предполагается дополнить вспомогательными элементами, такими как носимые вещи, для более интенсивной коммуникации умной системы с конечным пользователем.

Разработка подобной модели требует решения ряда технологических, инженерных, таких как оптимальная конфигурация системы, и программных задач. Программные задачи предполагают работу с открытыми графическими библиотеками [4]. В настоящий момент ведется работа по разработке методов локализации объектов в системах дополненной реальности в условиях неполных данных [5]. Нахождение необходимых объектов (маркеров) в видеопотоке является не тривиальной задачей, особенно в условиях низкого качества изображения – неполных данных. Это может быть обусловлено малой освещенностью, смазом, дефокусировкой, помехами, когда дистанционное зондирование пространства осуществляется в условиях чрезвычайных ситуаций, связано с высоким риском. Данные системы могут найти широкое применение не только в решении проблем людей с ограниченными возможностями, но и при чрезвычайных ситуациях и полевых работах.

Литература

1. Theory and applications of markerbased augmented reality [Электронный ресурс]. – Режим доступа: http://www.vtt.fi/inf/pdf/science/2012/S3.pdf (дата обращения: 04.01.2019).
2. Leibowitz H.M. et al. The Framingham Eye Study monograph: an ophthalmological and epidemiological study of cataract, glaucoma, diabetic retinopathy, macular degeneration, and visua acuity in a general population of 2631 adults, 1973–1975 // Survey of ophthalmology. – 1979. – V. 24. – P. 335–610.
3. Depth Map from Stereo Images [Электронный ресурс]. – Режим доступа: http://docs.opencv.org/master/dd/d53/tutorial_py_depthmap.html#gsc.tab=0 (дата обращения: 04.01.2019).
4. Открытая графическая библиотека [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/OpenGL (дата обращения: 04.01.2019).
5. Визильтер Ю.В. и др. Обработка и анализ изображений в задачах машинного зрения: Курс лекций и практических занятий. – М.: Физматкнига, 2010. – 672 с.

Васильченко Иван Дмитриевич
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41601
Направление подготовки: 09.04.02 – Компьютерная графика и веб-дизайн
e-mail: vasilchenko.6520@gmail.com

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004.51

ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ UI/UX-ДИЗАЙНА MHEALTH-ПРИЛОЖЕНИЙ

Васильченко И.Д.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

Мобильная медицина все больше внедряется в повседневную жизнь человека. Использование различных mHealth-приложений помогает пользователям следить за состоянием своего здоровья и поддерживать здоровый образ жизни. Очень важно, чтобы пользователям было удобно пользоваться mHealth-приложениями. От удобства и внешнего вида интерфейсов зависит эффективное взаимодействие пользователей с приложениями и желание пользоваться ими длительное время.
Ключевые слова: mHealth, приложение, интерфейс, проектирование, юзабилити.

За последние годы значительное развитие получила мобильная медицина, одно из направлений здравоохранения. Правительством Российской Федерации была составлена госпрограмма, направленная на развитие здравоохранения в России. В госпрограмме существует направление по развитию и внедрению инновационных методов диагностики, профилактики и лечения, а также основ персонализированной медицины. В госпрограмму инвестировано около 30 трлн руб.

Существует термин mobile health (mHealth), который означает использование мобильных устройств в медицинских целях: предоставление медицинской помощи; мониторинг различных параметров состояния здоровья пациента; предоставления медицинской информации врачам и пациентам; сбор клинических данных о состоянии здоровья пациента.

mHealth может улучшить качество и доступность медицинского обслуживания, упростит процесс диагностирования болезней и повысит общую информированность населения о здоровье. На сегодняшний день mHealth является одним из значимых направлений в развитии здравоохранения страны.

Недавние исследования мирового рынка приложений, которые проводились в 2016–2017 гг., показали, что сфера mHealth активно развивается, ежегодный рост количества приложений в среднем составляет 25%. Также с каждым годом растут доходы разработчиков. Аналитики утверждают, что общий уровень дохода рынка mHealth-приложений поднимется до 31 млрд долл. к 2020 году.

В 2017 году Россия заняла 6 место по объему выручки и реализации технологий мобильной медицины. Доход равнялся 800 млн долл., что составляло 3% от мировой прибыли рынка mHealth.

Весь рынок mHealth имеет два крупных сектора:

1. мобильные приложения, которые работают в связке с мобильными устройствами, или самостоятельно. Этот сектор сильно зависит от государственного финансирования;
2. мобильные устройства, которые используются в фитнесе и медицине. Этот сектор является финансово независимым.

Все mHealth-приложения можно разделить на несколько категорий [1]:

1. фитнес;
2. образ жизни;
3. питание;
4. болезни;
5. женское здоровье;
6. лечение;
7. справочники;
8. другое.

На рисунке представлено деление рынка mHealth-приложений. Большую часть рынка занимают категории фитнес и образ жизни. Среди населения наблюдается потребность в самостоятельном поддержании здоровья и физической формы. Меньшую долю рынка занимают категории болезни и лечение. Это связано с тем, что для их разработки нужно большое количество ресурсов.

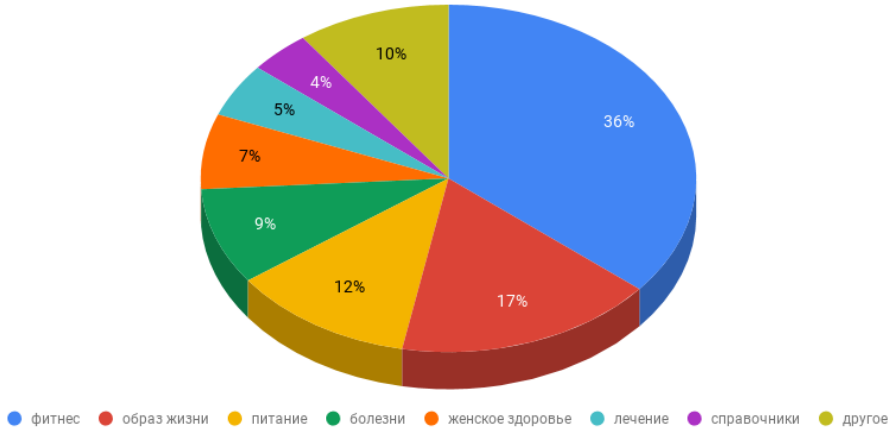


Рисунок. Деление рынка mHealth-приложений по категориям

Аналитики прогнозируют дальнейший рост количества пользователей mHealth-приложений. К 2020 году будет насчитываться 2,2 млрд человек, которые хотя бы раз скачивали mHealth-приложение. При этом 551 млн человек будут являться постоянными пользователями. Наиболее востребованными будут считаться приложения для хронических больных и для связи с врачом.

В последнее время люди все больше начали задумываться о собственном здоровье и физическом состоянии тела. Когда начала появляться потребность в приложениях, которые могут помочь контролировать общее состояние здоровья и отдельные его параметры, рынок быстро отреагировал и начал предлагать решения. На текущий момент выпущено большое количество приложений с разными назначениями [2]:

1. сбор статистики о физической активности пользователя;
2. анализ фаз сна;
3. измерение пульса;
4. трекинг бега;
5. составление индивидуальной программы тренировок;

6. прогнозы женского здоровья;
7. расчет потребления продуктов и составление рациона питания;
8. напоминание о приеме лекарств согласно предписаниям лечащего врача;
9. контроль потребления воды;
10. избавление от вредных привычек (курение).

При создании приложений очень важно уделить вниманию дизайну. Пользователям должно быть комфортно взаимодействовать с приложением. Удобство использования приложения сильно связано с уровнем лояльности пользователей.

Существует понятие юзабилити – это качественный показатель, определяющий эффективность и удовлетворенность пользователей при выполнении своих задач. Эффективность использования приложений определяется следующими показателями [3]:

1. соотношение между успешными и ошибочными действиями пользователей;
2. процент использования всего функционала приложения;
3. когнитивная нагрузка на пользователей;
4. легкость изучения нового функционала;
5. время выполнения определенных задач;
6. количество ошибок, совершенных пользователями, и время их устранения;
7. запоминаемость интерфейса;
8. субъективный эмоциональный аспект пользователей.

mHealth-приложения должны эффективно выполнять поставленные задачи и быть удобными для большего количества людей. Интерфейсы приложений должны быть простыми, понятными и внимательными к пользовательским ошибкам. Некоторые задачи, такие как поиск, сортировка, сводка, перекрестная проверка и вычисление должны выполняться в фоновом режиме для снижения когнитивной нагрузки на пользователя. Специалисты сформировали ряд правил, которые могут помочь улучшить юзабилити mHealth-приложений [4]:

1. правило «7±2». В памяти человека одновременно может находиться только 5–9 объектов, поэтому при проектировании навигации приложения не стоит создавать количество разделов больше этого числа;
2. правило «80/20». Принцип Парето говорит о том, что 20% действий приводит к 80% результатов. Примерно в 80% случаев будет использовано только 20% функций всего приложения, и при проектировании акцент нужно делать именно на самом важном функционале;
3. доступность информации. Самая важная информация должна быть доступна за минимальное количество действий. Также время загрузки информации не должно превышать 2 с;
4. очевидная навигация. Пользователь должен в любой момент времени понимать, в каком разделе он находится. Для этого можно добавить индикацию активного раздела и обозначить предыдущий экран;
5. понятные заголовки. Заголовки должны быть короткими, ясными и однозначно определять содержимое;
6. читаемый шрифт. Текст должен иметь достаточный размер и контрастность с фоном, чтобы его мог прочесть любой пользователь.

Несмотря на существующие правила, остается ряд проблем, связанных с юзабилити mHealth-приложений [5]:

1. беспроводные подключения. Подключение к носимым устройствам и синхронизация с ними происходит медленно и не всегда своевременно. Это влияет на производительность приложений, работающих в связке с носимыми устройствами, так как актуальные данные приходится ждать некоторое время;
2. ввод данных. Имеются различия в вводе данных на смартфоне и на компьютере с механической клавиатурой. При использовании маленького экрана смартфона уменьшается

скорость ввода данных и увеличивается вероятность совершить ошибку. Часто пользователи негативно относятся к сложным формам ввода. Если пользователи испытывают неудобство с интерфейсом, то они могут совсем прекратить пользоваться приложением;

3. вычислительные мощности. Смартфоны обладают низкой вычислительной мощностью, поэтому для обработки данных часто приходится отсылать их на сервер и получать результат обратно;
4. контекст использования. При использовании смартфонов пользователи часто переключают свое внимание между разными приложениями, уведомлениями, окружающими людьми и другими окружающими объектами. Поэтому при возвращении в приложение пользователи должны сразу понимать, на чем они остановились;
5. размеры экранов. Смартфоны обладают маленькими размерами экранами, поэтому при проектировании интерфейсов нужно учитывать, что в поле зрения может попасть не вся информация. Также графические элементы (диаграммы, графики, изображения) будут маленького размера и должны иметь возможность масштабироваться.

Таким образом, разработчики приложений придерживаются устоявшихся правил UI/UX-дизайна, которые помогают сделать приложение удобнее и привлекательнее для пользователей. Очень важно, чтобы пользователям было удобно пользоваться mHealth-приложениями. От удобства и внешнего вида интерфейсов зависит эффективное взаимодействие пользователей с приложениями и желание пользоваться ими длительное время.

Литература

1. Сошников С.С., Горкавенко Ф.В., Владимиров С.К., Ночёвки Е.В., Борисенко А.А., Котляр В.А., Фролкова А.Б. Классификация мобильных медицинских приложений, принципы и этические стандарты для их имплементации в клиническую практику // Медицинские технологии. Оценка и выбор. – 2017. – № 3(29). – С. 53–58.
2. Фокин С.Ю., Киричек Р.В. Обзор медицинских приложений, устройств и технологий связи Интернета Вещей // Информационные технологии и телекоммуникации. – 2016. – Т. 4. – № 4. – С. 67–80.
3. Бакаев М.А. Современные тенденции в автоматизированной оценке юзабилити и поведенческие факторы в алгоритмах поисковых систем // Программные продукты и системы. – 2017. – № 3. – С. 447–455.
4. Лайкова А.А. Юзабилити сайта: принципы и методы оценки // Актуальные проблемы авиации и космонавтики. – 2016. – № 12. – С. 745–747.
5. Mourouzis A., Chouvarda I., Maglaveras N. Mhealth: common usability and user experience practices and flaws // European, Mediterranean & Middle Eastern Conference on Information Systems. – 2015. – P. 18–21.

Волков Виктор Олегович
Год рождения: 1992
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4255
Направление подготовки: 09.04.01 – Системное администрирование
аппаратно-программных комплексов и сетей
e-mail: volkof92@mail.ru

Авксентьева Елена Юрьевна
Год рождения: 1981
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: avksentievaelena@rambler.ru

УДК 004.77

МОДЕРНИЗАЦИЯ ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ МОБИЛЬНОГО УЗЛА СВЯЗИ НА БАЗЕ АВТОМОБИЛЯ КАМАЗ

Волков В.О.
Научный руководитель – к.педагог.н., доцент Авксентьева Е.Ю.

Работа выполнена в рамках темы НИР № 615892 «Исследование и разработка в области информационных технологий».

В данной работе изложено предназначение мобильного узла связи, раскрываются его возможности, и представлены схема и оборудование, входящее в состав его сети, а также указываются недостатки, выявленные в ходе его эксплуатации.
Ключевые слова: мобильный узел связи, чрезвычайная ситуация, локальная вычислительная сеть, Интернет, Интранет.

Для эффективного управления силами и средствами спасательных формирований при ликвидации последствий чрезвычайных ситуаций (ЧС) используются средства связи, обеспечивающие надежную передачу оперативной информации из района ЧС в пункты управления [1, 2].

Для обеспечения связи не только в городских, но и в полевых условиях используется мобильный узел связи (МУС) на базе автомобиля КамАЗ с шасси высокой проходимости, который способен выполнять поставленные задачи независимо от места возникновения ЧС, погодных и климатических условий.

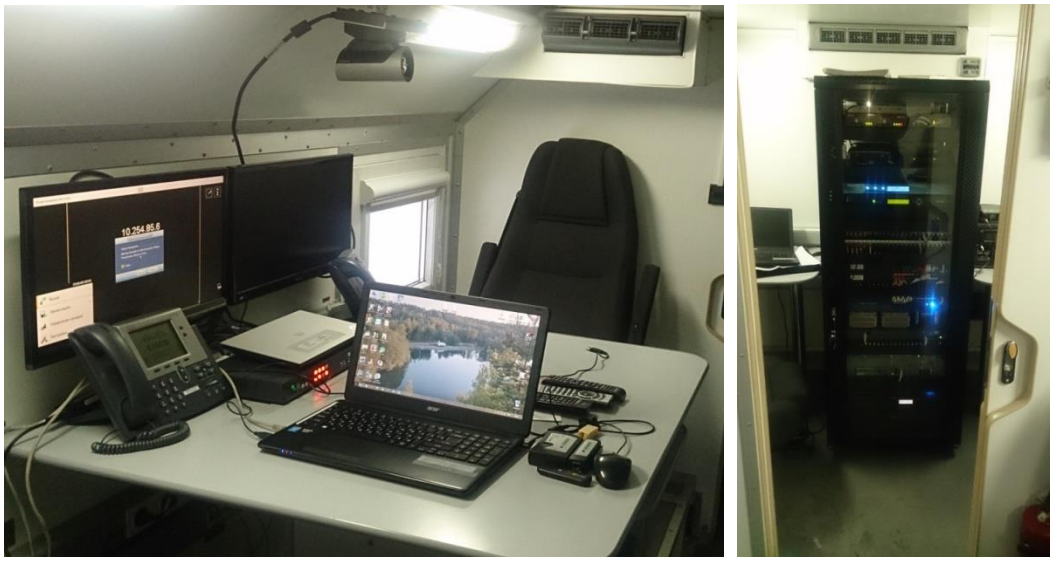
МУС предназначен для обеспечения устойчивой связью в полевых условиях оперативных групп подразделений МЧС России при ликвидации последствий ЧС природного и техногенного характера, а также:

– доставку личного состава оперативной группы к месту ЧС;
– развертывание средств оперативной группы на месте ЧС;
– обеспечения автономной работы сил и средств оперативной группы;
– обеспечение мультисервисными услугами связи;
– сбор, обработку и передачу различной информации о характере ЧС;
– визуализацию получаемой информации;

– управление силами и средствами, задействованными при проведении аварийно-спасательных работ;
– обеспечение личному составу оперативной группы комфортных условий для эффективной работы в ходе пребывания на месте ЧС.

Кунг МУС разбит на несколько отсеков, таких как: грузовой отсек для возимого имущества, имеющий собственную дверь; оперативный отсек, в котором размещаются не только рабочие места, но и два спальных места, а также бытовой шкаф, вмещающий в себя холодильник, СВЧ-печь, чайник и полки под личные вещи; связной отсек, в котором размещаются коммутационный шкаф и оборудование радиосвязи.

Оперативный отсек и отсек связи оборудованы пятью рабочими местами для операторов и специалистов связи, позволяющими пользоваться услугами Интернет и Интранет, IP-телефонией, осуществлять сеансы видеоконференцсвязи (ВКС), пользоваться средствами радиосвязи (рис. 1, а).



а б

Рис. 1. Вид рабочих мест в оперативном отсеке (а); коммутационный шкаф (б)

Также в связном отсеке располагается коммутационный шкаф, включающий в себя все основное коммутационное и каналообразующее оборудование, систему управления спутниковой станцией, источник бесперебойного питания (рис. 1, б).

Комплекс технических средств МУС включает в себя функциональные системы, а также специальное и вспомогательное оборудование:
– средства связи и передачи данных;
– оборудование внешних каналов связи и привязки операторов связи;
– средства вычислительной техники;
– аппаратуру криптозащиты;
– средства ВКС;
– систему видеонаблюдения;
– средства визуализации получаемой информации;
– оборудование приема спутникового телевидения и воспроизведения медиаконтента;
– систему электропитания;
– систему жизнеобеспечения.

Локальная вычислительная сеть (ЛВС) мобильного узла представляет собой сеть со смешанной топологией, которая объединяет в себя несколько подсетей типа «звезда». Таких «звезд» можно насчитать три. Это подсеть пользователей Интернет, подсеть пользователей Интранет, с доступом в ведомственную сеть, которая физически отделена от сети Интернет. А также «звезда», которая отвечает за ввод каналов связи в мобильный узел. В ней

располагаются такие устройства, как DSL-модем, принимающий сигнал от узла привязки на дистанции до 2 км по медной паре, радиорелейные станции, предоставляющие доступ к каналу связи на дистанции до 8 км в прямой видимости. Также имеется терминал спутниковой связи BGAN и малая спутниковая станция (VSAT) с диаметром зеркала 1,2 м (рис. 2).

Рис. 2. Мобильный узел с развернутой спутниковой станцией

В ходе эксплуатации МУС на протяжении более 18 месяцев были выявлены следующие недостатки и недоработки, требующие модернизации. К ним можно отнести:

1. необходимость размещения в связном лючке разъема для вывода видеоканала HDMI. В процессе работы на МУС была выявлена необходимость вывода видеосигнала по HDMI-кабеля из автомобиля и подача его на средства отображения, например, такие как оборудование ОКСИОН (Общероссийская комплексная система информирования и оповещения населения), представляющее собой передвижные (мобильные) или стационарные дисплеи для оповещения и информирования населения в случае возникновения ЧС;
2. необходимость замены антенны зенитного излучения (АЗИ). При работе с АЗИ в движении и на месте был выявлен ряд недостатков, существенно влияющих на устойчивость связи. К таким недостаткам следует отнести несоответствие диапазона рабочих частот антенны и применяемой с ней радиостанцией. Установленная антенна имеет диапазон рабочих частот от 3 до 12,5 МГц, а применяемая радиостанция диапазон частот на прием: от 0,03 до 29,99 МГц и на передачу: от 1,60 до 29,99 МГц. Таким образом, данная АЗИ не может обеспечить полноценную работу радиостанции на всем ее диапазоне рабочих частот;
3. неудобство использования связного лючка при использовании навесного оборудования на мачте МУС. В ходе эксплуатации навесного оборудования на мачте мобильного узла, такого как IP-видеокамеры, усилитель 3G/4G-сигнала операторов сотовой связи, передающей антенны радиорелейной станции и др., было выявлено неудобство подключения данного оборудования через разъемы связного лючка. При подключении через лючек, все его разъемы оказались заняты под навесное оборудование, вследствие чего становится невозможным вводить и выводить через лючек основные и дополнительные каналы связи. Также происходит и в обратной ситуации – при вводе и выводе каналов связи через связной лючек, становится невозможным подключение через него навесного оборудование мачты мобильного узла;
4. ухудшение скорости передачи данных и устойчивости канала связи при работе через сети 3G/4G в условиях ЧС. При работе МУС в ЧС в городской местности в условиях плотной застройки и, вследствие чего, в условиях отсутствия «прямой видимости» спутниковой станции мобильного узла и спутника связи на геостационарной орбите Земли, для

обеспечения канала связи применяется 3G/GSM-модуль, работающий в сетях операторов сотовой связи. Но существует проблема: при возникновении ЧС знакомые и родственники пострадавших и возможных пострадавших начинают активно обзванивать своих близких, из-за чего операторы сотовой связи не справляются с возрастающей в разы нагрузкой на сеть. Как итог, сотовая связь в районе ЧС практически отсутствует, что, в свою очередь, сказывается на устойчивости канала связи мобильного узла;

5. неудобство управления IP-видеокамерой при трансляции изображения с нее в видеоконференцию. При трансляции изображения из района ЧС через IP-видеокамеру в видеоконференцию необходимо управление камерой (наведение камеры на объекты, зуммирование). В данный момент такое управление осуществляется через IP-интерфейс или специальную программу по управлению видеокамерой. Но у такого способа существует недостаток, такой как отображение элементов интерфейса управления камерой в видеоконференцию, что мешает передаче изображения в полном объеме. Чтобы обойти данный недостаток в настоящее время применяется дополнительный ноутбук для управления видеокамерами, что вносит неудобства и отнимает одно рабочее место;
6. нет возможности работы в сетях Интранет и Интернет из кабины водителя в движении и на месте. В процессе работы на мобильном узле связи была выявлена необходимость работы в сетях Интернет и Интранет из кабины водителя. Возможность такой работы позволит получать и анализировать оперативную информацию старшему машины. Например, такие как: метеосводка, изменение маршрута движения мобильного узла, информация о районе ЧС и др. Также станет возможным использование IP-телефона из кабины водителя.

Предложения по модернизации МУС и устранению существующих недостатков будут изложены в рамках выпускной квалификационной работы «Модернизация локальной вычислительной сети мобильного узла связи на базе автомобиля КамАЗ».

Схема ЛВС мобильного узла связи

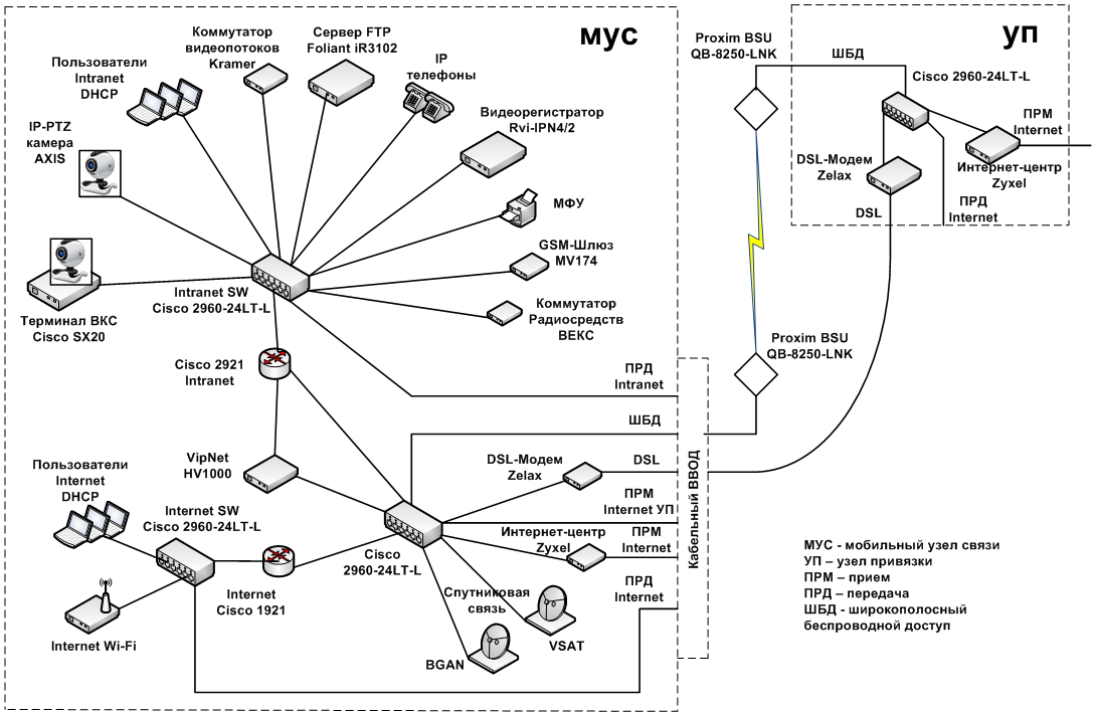


Рис. 3. Схема МУС

В данной работе было рассмотрено предназначение МУС, его возможности, конструктивное деление на отсеки, составлена схема его ЛВС с расположением основных

коммутирующих и периферийных устройств (рис. 3). Приведен перечень существующих недоработок, требующих модернизации.

Литература

1. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов. – 5-е изд. – СПб.: Питер, 2017. – 992 с.
2. Информационные технологии, связь и защита информации МЧС России // Информ. аналит. журн. – М.: МЧС России, 2015. – 217 с.

Воробьев Павел Валерьевич
Год рождения: 1997
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р3474
Направление подготовки: 44.03.04 – Компьютерная графика и дизайн
e-mail: passeridaepc@gmail.com

Махлай Дмитрий Олегович
Год рождения: 1990
Университет ИТМО, факультет программной инженерии
и компьютерной техники, ассистент
e-mail: teslandroid@gmail.com

Сопроненко Лариса Петровна
Год рождения: 1965
Университет ИТМО, факультет программной инженерии
и компьютерной техники, ст. преподаватель
e-mail: so_lar@mail.ru

Чернева Вероника Ивановна
Год рождения: 1989
Университет ИТМО, факультет программной инженерии
и компьютерной техники, ассистент
e-mail: chernnika@gmail.com

УДК 004.925

ИСПОЛЬЗОВАНИЕ ВИРТУАЛЬНОЙ РЕКОНСТРУКЦИИ ПРИ СОЗДАНИИ МУЛЬТИМЕДИЙНОЙ СИСТЕМЫ «РУССКИЙ СТИЛЬ»

Воробьев П.В., Махлай Д.О., Сопроненко Л.П., Чернева В.И.
Научный руководитель – Сопроненко Л.П.

Работа выполнена в рамках гранта РФФИ № 17-04-12034 «Мультимедийная информационная система «Архитектурно-художественный комплекс Феодоровский городок в Царском селе как пример Русского стиля».

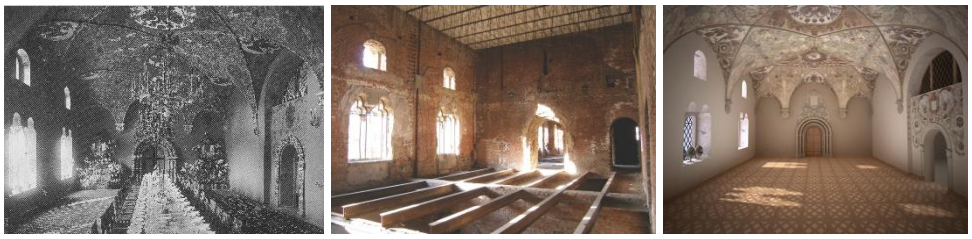
В работе рассмотрены результаты трехмерной реконструкции интерьеров Трапезной палаты архитектурного комплекса «Феодоровский городок», а также особенности воссоздания утраченных интерьеров. На примере реконструкции интерьеров Трапезной, Башенной Палаты и Притрапезной Церкви рассмотрены различные подходы в работе с утраченными элементами интерьера.
Ключевые слова: Феодоровский городок, цифровое культурное наследие, виртуальная реконструкция.

Феодоровский городок в Царском селе – уникальный пример комплекса, созданного в русском стиле. Автором проекта стал С. Кричинский, над росписями работали такие художники, как Н. Рерих, М. Васнецов и др. Наиболее репрезентативное здание комплекса – Трапезная палата [1–4].

Одной из наиболее интересных задач при виртуальной реконструкции Трапезной палаты стала реконструкция интерьеров. В зависимости от степени физической сохранности интерьера, от наличия или отсутствия исходных материалов (фотофиксации, графической фиксации, прорисей, архитектурных обмеров), послуживших основанием для виртуальной реконструкции, можно выделить несколько типов приемов виртуального воссоздания интерьеров.

– Трапезная палата – полностью утраченный интерьер, за исключением стен с проемами, частично измененными.
– Башенная палата – сохранилась общая геометрия сводов, фрагментарно – росписи, имеются акварельные зарисовки, факсимильные копии (акварель, гуашь, темпера), выполненные студентами мастерской реставрации церковной живописи Академии художеств под руководством проф. Крылова, а также фотографии гербов царств.
– Притрапезная церковь Сергия Радонежского – сохранился объем с повышенной и пониженной частями, а также эскизные варианты росписей, созданные по аналогам, в частности росписям Дионисия.

Наиболее репрезентативный и наименее сохранившийся интерьер – Трапезная (рис. 1). В качестве референсов были использованы единственная сохранившаяся черно-белая фотография интерьера 1917 года, прориси Царскосельской иконописной мастерской Г. Майоровой. В 2015 году Е. Акулаевой была воссоздана геометрия сводов и создана пространственно корректная 3D-модель интерьера.



а б в

Рис. 1. Состояние Трапезной: 1917 г. (а); 2010 г. (б); 3D-реконструкция (в)

В 2016 году студентами мастерской реставрации церковной живописи Академии художеств были созданы эскизные варианты цветового решения росписей сводов. Варианты студентов Хабарова, Шувалова и Сухорукова в виде текстур наложены на смоделированную поверхность сводов (рис. 2).

а б в г

Рис. 2. Реконструкция цветового решения потолка: Е. Акулаевой (а); Р. Шувалова (б); В. Сухорукова (в); А. Хабарова (г)

В 2018 году в процессе работы над созданием пространства виртуального музея русского стиля были смоделированы детали интерьера – обрамление дверей, дверные полотна, мебель (по сохранившимся акварельным эскизам начала XX века, дизайн разрабатывался Н. Бартрамом специально для интерьеров Трапезной палаты).

В работе по созданию виртуального интерьера Притрапезной церкви наиболее сложной задачей стало воссоздание колорита в пространстве, поскольку в эскизах росписей, созданных в 2009–2010 гг. студентами мастерской проф. Крылова: Андреенковой, Денисовой, Зацемирной, Родиковым, Шириней, не учитывалась разница в высотах частей помещения и направленность света (рис. 3).

Рис. 3. Референс для создания текстур Притрапезной Церкви – эскизы росписей мастерской Крылова, 2009–2010 гг.

При реконструкции интерьера Башенной палаты были использованы как материалы фотофиксации, так и факсимильные копии росписей потолка, и фиксационные акварельные зарисовки М. Капралова, представляющие собой гербы царств, входивших в Российскую империю. На примере реконструкции герба Сибирского царства можно оценить преимущества векторного изображения по отношению к растровому.

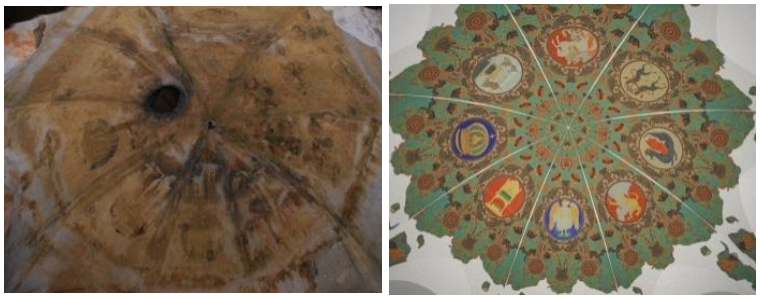
В создаваемой мультимедийной системе «Русский стиль» важное место отводится виртуальному музею, экспонатами которого, в частности, являются собственно пространства интерьеров здания Трапезной палаты с росписями сводов.

Для демонстрации реконструированных интерьеров выбрано два подхода:

1. интеграция демонстрационной версии на страницу сайта russianstyle.ifmo.ru при помощи технологии WebGL;
2. виртуальная прогулка при помощи гарнитуры виртуальной реальности с полным «погружением» в пространство.

Например, используя интерактивные элементы можно выбрать и посмотреть различные варианты цветового решения интерьера Трапезной.

В Башенной палате воссоздано пространство интерьера в целом, а также элементы интерьера – двери, пол и росписи потолка (рис. 4). При реконструкции росписей наибольшей проблемой стала невозможность использовать ни один из видов исходных материалов для качественного текстурирования, поскольку фотофиксация проводилась с целью создания художественных эскизов, без учета требований к освещению и допустимому искажению, а провести корректную фотосъемку нет возможности, так как объект недоступен.



а б

Рис. 4. Свод Башенной Палаты: состояние в 2010 г. (а); результат 3D-конструкции (б)

Акварельные схематичные развертки дают общее представление о колорите и сюжете росписей, в то время как взаимное расположение фрагментов зачастую недостоверно пространственно. Для профессиональной реконструкции росписей в 3D правильнее использовать оригиналы или факсимильные копии, выполненные, в частности студентами

мастерской проф. Крылова, однако к 2017 году сохранились не все фрагменты факсимильных копий.

Чтобы воссоздать утраченные фрагменты без потери качества, было принято нестандартное решение: векторная реконструкция контуров изображения с заполнением бесшовной текстурой штукатурки. На примере реконструкции герба Сибирского царства можно оценить преимущества векторного изображения по отношению к растровому.

Векторный метод реставрации текстур позволяет воссоздать фреску без потери качества, с возможностью быстрой корректировки форм и размеров элементов. Этот способ максимально удобен для точной и детальной компьютерной реконструкции.

При реконструкции интерьера Притрапезной церкви Сергия Радонежского было воссоздано уникальное пространство, сочетающее в себе плоский потолок над алтарной частью и высокое «небо» над помещением для молящихся (рис. 5). Эскизы росписей сюжетно соответствуют православным канонам и воспроизводят колорит, характерный для фресок Дионисия.

а б в

Рис. 5. Притрапезная Церковь: в 2010 г. (а); результат 3D-конструкции (б, в)

Дальнейшим развитием работы будет виртуальная реконструкция помещений лестницы Круга солнечного, Царской Палаты и др., а также использование полученных моделей при создании виртуального музея Русского стиля.

Литература

1. Sopronenko L.P., Akulaeva E.A. The virtual reconstruction of the interior of Refectory Feodorovsky town in Tsarskoye Selo // Electronic Imaging & the Visual Arts, IET. – 2015. – P. 119–121.
2. Smolin A., Sopronenko L., Akulaeva E., Lavrov A., Zvyagin K. Russian style through time. Experience of virtual reconstruction of the interior of the refectory of the Fedorovsky Gorodok in Tsarskoye Selo // 3rd International Multidisciplinary Scientific Conference on Social Sciences and Arts. – 2016. – V. 2. – № 4. – P. 571–578.
3. Sopronenko L.P., Korpan L.M., Lavrov A.V., Balkanskii A.A. Russian style: from creation to virtual reconstruction. On the centenary of Fedorovsky gorodok in Tsarskoye Selo // 4rd International Multidisciplinary Scientific Conference on Social Sciences and Arts, SGEM 2017. – 2017. – V. 1. – № 5. – P. 261–268.
4. Smolin A., Sopronenko L., Borisov N., Lavrov A., Melnikov V. Activities in the field of digital humanities based on creation an information multimedia portal architecture and art complex feodorovsky town in tsarskoye selo as an example of russian style // 5th International Multidisciplinary Scientific Conference on Social Sciences and Arts, SGEM 2018. – 2018. – V. 5. – № 6.1. – P. 315–320.

Смолин Артем Александрович
Год рождения: 1977
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.философ.н.
e-mail: smolin@corp.ifmo.ru

Габдуллазянова Зарина Рустемовна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4172
Направление подготовки: 09.04.02 – Дизайн человеко-компьютерных систем
e-mail: zarina.gabdullazianova@gmail.com

УДК 004.582

ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСОВ ДЛЯ ОБУЧЕНИЯ ДЕТЕЙ С ДИСЛЕКСИЕЙ

Габдуллазянова З.Р.,
Научный руководитель – к.философ.н. Смолин А.А.

В работе рассмотрено применение псевдозерновой культуры киноа в качестве нетрадиционного сырьевого ингредиента для российского потребителя с целью создания многокомпонентных рецептур, что обосновано ее высокой пищевой ценностью и содержанием белка отличительного качества для рационализации использования традиционных сырьевых ресурсов с целью разработки сбалансированных продуктов питания. В работе произведен обзор научных статей про дислексию, имеющихся научных исследований и наличия гайдлайнов по данной тематике. Были обозначены трудности данной темы, выявлена проблема отсутствия интерфейсов, специализированных под людей с дислексией. Описаны имеющие средства и пути решения данной проблемы.
Ключевые слова: дислексия, гайдлайн, дети, исследование, чтение, ассоциация, дислексики.

Дислексия, заболевание, находящееся на грани нейрологии, офтальмологии и психиатрии, характеризуется нарушением у детей освоения навыков чтения, что приводит к существенной задержке психического развития. У взрослых также возможно возникновение дислексии при травмах и опухолях головного мозга, других органических поражения центральной нервной системы [1].

Признаки дислексии обычно проявляются, когда ребенок начинает ходить в школу и начинает больше концентрироваться на обучении чтению и письму.

Человек с дислексией обычно:

– медленно читает и пишет;
– путает порядок слов в предложении;
– с легкостью воспринимать информацию в устной форме, но иметь трудности с уже записанной информацией;
– испытывать трудности с планированием и организацией;
– но в то же время, люди с дислексией имеют хорошие навыки в таких областях как творческое мышление и решение проблем.

Международная ассоциация по дислексии, как и большая часть литературы по психологии, определяет дислексию как специфическую неспособность к обучению [2–4], в то время как в некоторых исследованиях когнитивной неврологии дислексия упоминается как специфическая неспособность к чтению [5]. Однако в стандартных определениях Международной статистической классификации болезней (ICD-10) [6] и Диагностического и

статистического руководства по психическим расстройствам (DSM-IV) [6] дислексия указана как расстройство чтения и орфографии (ICD-10) или расстройство чтения и расстройство письменного выражения (DSM-IV). В испанской литературе дислексия почти никогда не упоминается как инвалидность, а как трудность или расстройство в отношении обучения и чтения [7]. Первая проблема заключается в том, где поставить дислексию в WACG 2.0 [8], поскольку неясно, является ли дислексия когнитивной инвалидностью.

В научных исследованиях количество дислексиков во всем мире варьируется от 3–10% [8], 15% в [7] от всего населения Земли, но довольно сложно быть уверенным в этих числах. Трудности с определением количества людей с дислексией связаны с природой заболевания.

Причины сложности выявления дислексии:

– проявления дислексии зависят от языка;
– трудная диагностика;
– сочетает в себе несколько заболеваний.

Проявления дислексии возможны только на языках с системами письма. В этом мире насчитывается около 6 900 языков, и около трети из них – оценки различаются – до сих пор не имеют систем письма. И, во-вторых, среди языков, имеющих системы письма, даже если исследования согласны с неврологической универсальностью дислексии, ее проявления различны для разных языков, что делает его обнаружение еще более сложным [9].

Несмотря на сложности относительно самого заболевания, существует официальный гайдлайн по графическому оформлению для дислексиков the Web Content Accessibility Guidelines 2.0/WCAG 2.0 [10]. В нем описывается работа с текстом и контрастом (рис. 1).

Text and background color	Color difference	Brightness difference
black (000000) & white (FFFFFF)	765	255
black (000000) & yellow (FFFF00)	510	226
black (000000) & creme (FAFAC8)	700	244
off-black (0A0A0A) & off-white (FFFFE5)	735	245
blue (00007D) & white (FFFFFF)>	640	241
dark brown (1E1E00) & light green (B9B900)>	310	137
brown (282800) & dark green (A0A000)>	240	107
blue (00007D) & yellow (FFFF00)	635	212

Рис. 1. Таблица контрастов [10]

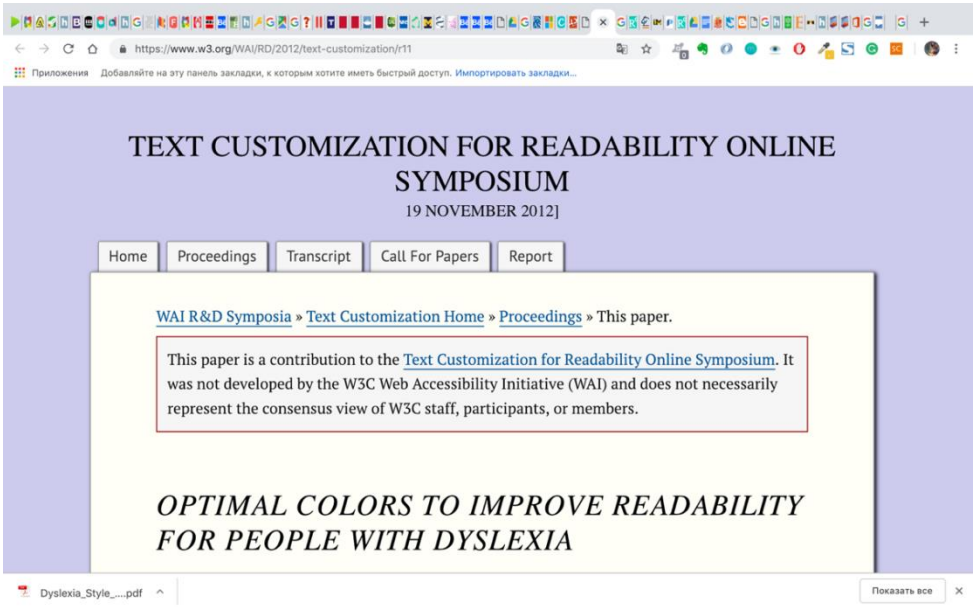


Рис. 2. Дата публикации WCAG 2.0 [10]

Данный гайдлайн может стать хорошей основой для разработки графических интерфейсов, но он является устаревшим и на сегодняшний день не содержит актуальной информации. Взаимодействие человека с компьютерными системами меняется чуть ли не каждый день, поэтому гайдлайн 2012 года публикации (рис. 2) вряд ли можно отнести к материалу, на который стоит равняться.

В России дислексия не так давно получило свое определение как заболевание, и на сегодняшний день существует «Ассоциация родителей и детей с дислексией». Ассоциация была основана в 2016 году.

Миссия Ассоциации: дать детям с дислексией и другими трудностями обучения возможность получить качественное фундаментальное образование и реализовать свой профессиональный потенциал.

Стратегией ассоциации являются: усовершенствование системы российского образования путем внедрения современных научных знаний и лучших отечественных и зарубежных практик по поддержке детей с трудностями обучения, законодательное закрепление прав таких учеников [11].

Планируется сотрудничество с данной ассоциацией для создания интерфейсов, которые помогут проводить тестирование для детей-дислексиков. При удачной попытке проведения данного исследования рассматривается возможность введения такого интерфейса в учебный процесс и специальную программу для детей с дислексией.

Разработка интерфейсов для дислексиков необходима, поскольку позволит им лучше социализироваться в современном обществе.

В дальнейшем ходе исследовательской работы планируется исследовать закономерности взаимодействия дислексиков с интерфейсами, создание методов проектирования интерфейсов, тестирование методов.

Литература

1. Французские биологи открыли возможную причину возникновения дислексии у человека, ТАСС [Электронный ресурс]. – Режим доступа: https://tass.ru/nauka/4656098/ (дата обращения: 04.01.2019).
2. Definition of dyslexia [Электронный ресурс]. – Режим доступа: https://dyslexiaida.org/definition-of-dyslexia/ (дата обращения: 04.01.2019).
3. Lyon G.R. Toward a definition of dyslexia // Annals of Dyslexia. – 1995. – V. 45(1). – P. 3–27.
4. Lyon G.R., Shaywitz S.E., Shaywitz B.A. A definition of dyslexia // Annals of Dyslexia. – 2003. – V. 53(1). – P. 1–14.
5. Vellutino F.R., Fletcher J.M., Snowling M.J., Scanlon D.M. Specific reading disability (dyslexia): What have we learned in the past four decades? // Journal of Child Psychology and Psychiatry. – 2004. – V. 45(1). – P. 2–40.
6. World Health Organization. International statistical classification of diseases, injuries and causes of death (ICD-10). World Health Organization, Geneva. – 10th edition. – 1993.
7. Carrillo M.S., Alegr´ıa J., Miranda P., S´anchez N. Evaluaci´onde la dyslexia en la escuela primaria: Prevalencia en espa˜nol (Evaluation of dyslexia in primary school: The prevalence in Spanish) // Escritos de Psicolog´ıa (Psychology Writings). – 2011. – V. 4(2). – P. 35–44.
8. Caldwell B., Cooper M., Reid L.G., Vanderheiden G. Web content accessibility guidelines (WCAG) 2.0. WWW Consortium (W3C). – 2008.
9. Mc Cathy J.E. and Swearengin S.J. What we know about dyslexia and web accessibility: a research review // Universal Access in the Information Society. – 2010. – V. 9. – P. 147–152.
10. Text customization for readability online symposium [Электронный ресурс]. – Режим доступа: https://www.w3.org/WAI/RD/2012/text-customization/r11 (дата обращения: 04.01.2019).
11. Ассоциация родителей и детей с дислексией [Электронный ресурс]. – Режим доступа: https://dyslexiarf.com/missiya-i-deyatelnost/ (дата обращения: 04.01.2019).

Гарифуллина Гузель Раифовна
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41622
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: giselegarifullina@gmail.com

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.25

ОПТИМИЗАЦИЯ РАБОТЫ ВЕБ-ПРИЛОЖЕНИЙ С ПОМОЩЬЮ УПРАВЛЕНИЯ КЭШИРОВАНИЕМ

Гарифуллина Г.Р.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

В работе представлено комплексное введение в проблему кэширования в контексте веб-приложений, с основной ориентацией на решения, влияющие на улучшения пользовательского опыта с сайтом. Данная работа может служить в качестве отправной точки для разработчиков, которые хотят реализовать кэширование в веб-приложении.
Ключевые слова: веб-приложения, кэширование, клиентская часть, HTTP-заголовки, Service Worker.

Веб-приложение – это компьютерная программа, состоящая из двух частей: клиента и сервера, которая предназначена для работы в веб-браузере. Обе части веб-приложения работают параллельно, независимо и общаются между собой через HTTP-запросы:

1. клиентская часть находится в браузере и отвечает за пользовательский интерфейс;
2. серверная часть содержит в себе бизнес-логику веб-приложения и отвечает на HTTP-запросы клиента.

Веб-браузер выступает в качестве универсального клиента для любого веб-приложения. Во время сеанса веб-браузер запрашивает необходимые страницы у сервера. Сервер посылает запрашиваемые ресурсы клиенту. В дальнейшем браузер их интерпретирует и отображает пользователю.

Кэширование – это процесс сохранения результата долгих вычислений или операций для дальнейшего использования. Кэширование позволяет ускорить получение данных, вычисление сложных операций и уменьшить нагрузку на источник.

Кэширование может использоваться в различных частях сети. Например, в браузере, на сервере, в базе данных, на прокси-серверах, в обратных прокси-серверах, на шлюзах и т.д. На рисунке показаны основные локации кэша в веб-приложении.

Различные локации отвечают за кэширование различных ресурсов и служат для различных целей. Например, кэширование на сервере используется для улучшения производительности и масштабируемости сайта, в то время как кэширование на клиенте – для ускорения времени загрузки страницы и уменьшения сетевого трафика [1].

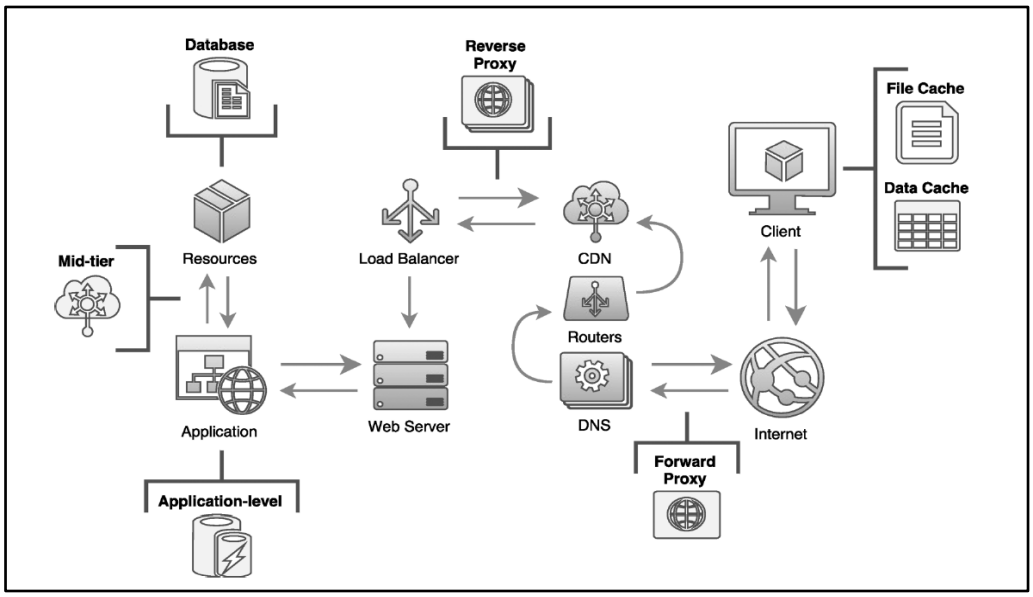


Рисунок. Возможные локации кэша в веб-приложении

Как правило, коэффициенты попадания кэша бывают более высокими, если кэш находится ближе к источнику информации (например, на оборудовании или базы данных), в то время как, чем ближе кэш к месту использования, тем большую выгоду он дает в случае попадания, даже несмотря на более низкие коэффициенты попадания [2].

В ходе данной работы интерес представляло кэширование в браузере, так как он находится ближе к клиенту, следовательно, дает больший выигрыш для пользователя. Например, исследование [3] показало, что доступ к сайту при полном кэшировании браузера может быть в 2,5 раза быстрее, чем при его отсутствии.

Есть два способа кэширования в браузере:

1. HTTP-заголовки;
2. Service Workers.

При использовании кэширования с помощью HTTP-заголовков при отправке контента сервер добавляет в HTTP-заголовок ответа директивы, контролирующие кэширование.

Данные директивы можно разделить на три категории:

1. определяющие ограничения на кэширование (no-store, public, private, no-cache);
2. определяющие время, в течение которого контент считается свежим (Expires, max-age, s-maxage);
3. валидаторы – HTTP-заголовки, помогающие определить текущую версию контента. Они используются для проверки актуальности данных. Главное преимущество использования валидаторов – это отсутствие необходимости повторной загрузки контента в кэш, если данные на сервере не изменились (Last-Modified, ETag).

Вместе свежесть и валидация кэша являются наиболее важными способами работы кэша с контентом. Свежее представление будет немедленно доступно из кэша, в то время как проверенное представление позволит избежать повторной загрузки всего представления, если оно не изменилось.

Service Worker – это скрипт на JavaScript, который браузер запускает в фоновом режиме отдельно от главного потока веб-страницы. Service Worker работает независимо от веб-приложения, с которым он связан, и у него есть свой жизненный цикл.

Service Worker может использоваться для:

1. кэширования ресурсов;
2. доступа к сайту в оффлайн режиме;
3. периодической фоновой синхронизации;
4. всплывающих уведомлений.

Помимо добавления новой функциональности в веб-приложение, использование Service Worker имеет множество преимуществ. Например, как показало исследование IOWA [4] добавление Service Worker к сайтам, использующих HTTP-заголовки, улучшает производительность сайта:

1. в среднем страницы, использующие Service Worker, загружаются быстрее как для новых, так и для повторных посетителей;
2. загрузка страниц, использующих Service Worker, происходит почти мгновенно;
3. Service Worker, если не активен, то требует время для запуска. Тем не менее, наличие неактивного Service Worker лучше, чем его отсутствие.

Поэтому есть смысл совмещать оба метода.

Таким образом, на основе проведенного анализа можно сделать вывод, что использование кэширования в веб-приложениях дает ряд преимуществ: повышает производительность сайта, уменьшает время загрузки, сокращает время отклика и уменьшает сетевой трафик. При этом в зависимости от выбора локации кэша меняются цели кэширования. Для улучшения пользовательского опыта в основном используется кэширование в браузере. В ходе данной работы были представлены и подробно описаны основные подходы к кэшированию на клиентской стороне веб-приложения.

Однако использование кэширования имеет ряд недостатков: дополнительный расход памяти, дополнительные вычислительные затраты на выявление устаревшего кэша и подгрузки нового при необходимости, риск использования устаревших данных, что может привести к некорректному поведению приложения или снизить его производительность. В связи с вышеизложенными причинами можно сделать вывод, что выбор правильных данных и операций для кэширования является одной из основополагающих задач при разработке веб-приложения.

Литература

1. Mohan C. Caching Technologies for Web Applications // VLDB. – 2009. – V. 1. – P. 726.
2. Mertz J., Nunes I. Understanding application-level caching in web applications: a comprehensive introduction and survey of state-of-the-art approaches // ACM Computing Surveys (CSUR). – 2018. – V. 50(6). – P. 98:1–98:34.
3. Performance Research, Part 2: Browser Cache Usage [Электронный ресурс]. – Режим доступа: https://yuiblog.com/blog/2007/01/04/performance-research-part-2/ (дата обращения: 04.01.2019).
4. Measuring the Real-world Performance Impact of Service Workers [Электронный ресурс]. – Режим доступа: https://developers.google.com/web/showcase/2016/service-worker-perf/ (дата обращения: 04.01.2019).

Геворкян Аделина Арменовна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4270
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: geborkyan.adelina@yandex.ru

Меженин Александр Владимирович
Год рождения: 1959
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: mejenin@mail.ru

УДК 004.94

МЕТОДЫ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПОИСКА ЛОКАЦИЙ

Геворкян А.А., Меженин А.В.
Научный руководитель – к.т.н., доцент Меженин А.В.

В ходе работы были рассмотрены возможности для поиска локаций съемок кинофильмов, а также проведены эксперименты для проверки эффективности одного из методов поиска.
Ключевые слова: локейшн-менеджмент, viewfinder, веб-приложение.

Пре-продакшн фильма – это первый подготовительный этап сложного процесса съемки фильма. Конец этого этапа означает, что сценарий уже написан, актеры подобраны, костюмы готовы, набрана необходимая техника и подготовлен список локаций для фильма. Локация для съемки является важным элементом кинопроизводства не только потому, что именно в ней снимается кино, она также обеспечивает атмосферу фильма, задает настроение зрителю, позволяя погрузиться в жизнь персонажей [1–5].

Поиск локаций является важной частью процесса пре-продакшна и достаточно непростой. Есть множество требований, которым должна соответствовать любая локация. Такие характеристики как: местоположение солнца в момент съемок, наличие электропитания, доступность локации, присутствие ненужного шума, являются важными проблемами съемок, которые необходимо заранее решать.

Одной из востребованных профессий в нынешней киноиндустрии является локейшн-менеджер. Это человек, занимающийся поиском и подготовкой места съемок для фильма или телепередачи, получающий от режиссера или художника-постановщика примерное описание натуры или объекта, необходимого для съемок. Иначе говоря, это директор по локациям.

У локейшн-менеджера есть ряд задач и проблем, которые необходимо учитывать при поиске локаций. Во-первых, не каждая локация может быть доступна для съемок. Если для сцены подобраны объекты государственные или принадлежащие кому-либо, то необходимо просить разрешение на съемку у собственников, зачастую это стоит денег. Во-вторых, свет в локации является важным фактором. В зависимости от направления солнца или погодных условий локация может либо соответствовать сцене, либо нет.

Также нужно учитывать наличие электропитания в локации, необходимость генератора, наличие постороннего шума в виде железной дороги или аэропорта. Локация может не всегда до конца соответствовать требованиям режиссера, но важно понимать, что ее можно дополнить декорациями, поэтому также нужно обратить внимание на возможность

постановки каких-либо декораций. Учитывать расстояние от киностудии и возможность перевозки всей съемочной группы до места съемок – тоже важный фактор. Зачастую выбираются места, до которых проще добираться. В ином случае необходимо продумать маршрут, и иметь возможность для поездки в другой город или другую страну.

Локейшн-менеджмент является достаточно востребованной профессией, так как натурные съемки, несмотря на серьезное развитие компьютерной графики, все равно являются популярными и необходимыми. Особенно в бюджетном кино, небольшая киностудия или студенческая команда вряд ли сможет обеспечить качественную компьютерную обработку сцены, поэтому в таком кино зачастую используются исключительно натурные локации.

Одним из простых и недорогих способов поиска натурной локации в последнее время можно называть поиск локации с приложением с функцией Viewfinder (рис. 1). Viewfinder, или на русском «видоискатель» предназначен для отображения кадра, снимаемого на саму камеру. Они бывают оптическими и электронными.

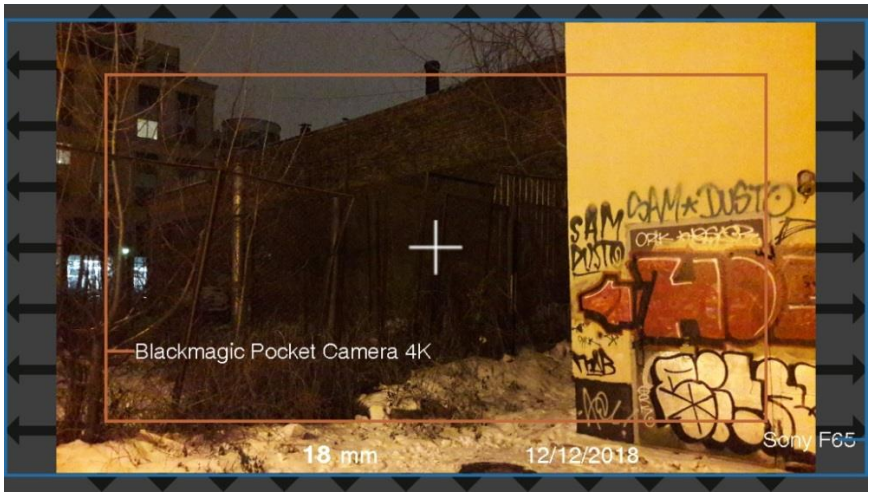


Рис. 1. Приложение с функцией Viewfinder

Оптический видоискатель представляет собой небольшое отверстие, которое показывает изображение, захватываемое объективом и при этом, имеет так называемую «точку стрельбы», которая помогает сфокусироваться на изображении.

Электронный видоискатель является более сложной технологией. Он позволяет увидеть гораздо больше, так как представляет собой небольшой экран, который показывает, что снимается камерой в данную секунду.

Сейчас становится довольно популярным метод поиска локаций с приложением, моделирующим Viewfinder-камеры. Из основных и очевидных плюсов такого метода – это возможность не носить с собой огромное количество техники на локацию и выбирать, какая оптика больше подойдет для съемки. Это экономит время и силы. При этом такие приложения могут помочь сузить выбор техники для съемок. А если говорить о малобюджетном кино, то при использовании таких приложений можно четко понимать, какую технику нужно использовать и брать в аренду. Само собой у малобюджетного кино не всегда есть возможность покупать профессиональные камеры.

Еще одним помощником в поиске локаций является всемирная сеть. А точнее сайты и сервера, созданные специально для тех кинокомпаний, которые пренебрегают помощью локейшн-менеджера ввиду необходимости оплаты таких услуг.

Большинство сайтов для поиска локаций имеют: одинаковую структуру, возможность ввести название локации в поисковую строку, увидеть варианты локаций по запросу, выбрать одну из них и узнать информацию о ее местоположении, и в случае, если это место принадлежит какой-либо компании, увидеть телефон и почту собственника, с которым можно договориться о проведении съемок.

Метод поиска локации на сайте является бюджетным, за счет того, что идет экономия на человеке, который сам договориться за съемочную группу. Съемочная группа решает вопрос самостоятельно. Однако не всегда такие сайты могут предоставить информацию в большом объеме.

Во-первых, на некоторых сайтах указано достаточно небольшое количество информации о местоположении локации, бывает, что это просто почта и телефон, привязанные к месту. Где-то отсутствуют фотографии, и не на каждом сайте можно предложить свою локацию.

Изучив интерфейсы сайтов и информацию о них, было принято решение в ходе данного исследования о необходимости создания сервера с базой данных локаций, используемого как кроссплатформенное веб-приложение.

Веб-приложение – это сайт с элементами интерактива, т.е. пользователь может взаимодействовать с материалом, функциями сайта, загружать информацию.

В ходе работы планируется создать работающий прототип веб-приложения для поиска локаций фильмов. Основная задача веб-приложения: предоставить пользователям доступ к базе данных локаций для съемок кинофильмов с возможной загрузкой найденных локаций для пополнения базы, т.е. приложение, уже в себе содержащее определенную базу данных, пополняемую самими пользователями.

На данный момент созданы уже первые графические прототипы приложения и продуман его функционал. Пользователь сможет зайти на сайт, ввести в поисковик название локации, получить варианты локаций по названию и увидеть сразу же информацию об их местоположении, рейтинге и даже информацию о том, снималось ли что-нибудь в этом месте ранее. При выборе локации, можно посмотреть фотографии, снятые в этом месте, причем просмотр фотографии осуществляется через функцию Viewfinder. Пользователь наводит мышь на интерактивную линейку и может прокрутить колесиком миллиметражи камер. Стандартом заданы две рамки 16:9, меняя миллиметраж можно сравнивать захват изображения двух камер и выбирать, какая камера подойдет больше.

Такая функция Viewfinder может быть помощником в выборе камеры, а также в создании интерактивности web-приложения. В ходе работы было проведено статистическое исследование об используемости приложений с функцией Viewfinder. Тридцать респондентов, учащиеся киновузов и уже закончившие режиссеры и операторы искали локацию по заданию с приложением с видоискателем и без него. Было дано задание – найти локацию для сцены, где двое мужчин осуществляют передачу нелегального оружия.

Рис. 2. Один из результатов эксперимента – найденная локация

Поиск проводился во дворах Петербурга, каждому респонденту предоставлялась возможность найти локацию самостоятельно и с помощью приложения с функцией Viewfinder. Во избежание эффекта последовательности, череда использования приложения менялась, т.е. некоторые респонденты начинали искать локации с приложениями, затем

искали без него, и наоборот. На секундомере засекалось время, за которое респондент найдет локацию (рис. 2).

Было найдено шестьдесят локаций, получено две выборки, первая показывает время, за которое респонденты искали локации без приложения, вторая – с приложением. Выборки считаются парными, так как в эксперименте задействована одна группа людей из 30 человек. Для данной работы был выбран критерий W-Уилкоксона, так как выборки отвечают его требованиям. Был проведен статистический расчет. И правосторонняя проверка значимости нацеленная на то, что с приложением локации искали быстрее.

Формула Уилкоксона для получения статистического критерия:

$$W = \sum_{i=1}^{n}[\text{sign}(x_i - y_i) \cdot R([\,x_i - y_i])],$$

где $R([x_i - y_i])$ – это ранг разности первой и второй выборки; $\text{sign}(x_i - y_i)$ – это знак ранжированных разностей, т.е. W равняется сумме всех рангов разностей между первой и второй выборкой.

Так как выборка была достаточно большой, критерий Уилкоксона преобразовывался в z-критерий, и значимость считалась по нормальному распределению. Формула z-критерия:

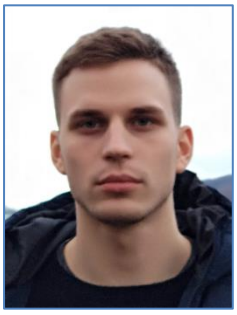
$$z = \frac{W+0{,}5}{\sqrt{\frac{n(n+1)(2n+1)}{6}}}.$$

В итоге полученный результат z=–1,835724287 имеет значимость 3,32% – результат является статистически значимым по критическим значениям z при правосторонней проверке гипотезы с уровнем значимости 5%.

Это говорит о том, что с приложением с функцией Viewfinder локации искались быстрее, а значит, подобное дополнение на странице web-приложения, поможет усилить эффективность поиска с помощью этого web-приложения. И само web-приложение будет очень полезным для киносоздателей, как инструмент для поиска локаций и примерного выбора оптики для съемки в локации.

Литература

1. Чемонин Р. Локейшн-менеджер: Роман Кантария [Электронный ресурс]. – Режим доступа: https://tvkinoradio.ru/article/article6469-lokejshn-menedzher-roman-kantariya (дата обращения: 05.01.2019).
2. Gladstone S. Location Scouting for Filmmaking: A Basic Guide [Электронный ресурс]. – Режим доступа: https://www.bhphotovideo.com/explora/video/tips-and-solutions/location-scouting-basic-guide (дата обращения: 25.12.2018).
3. Lim R. 7 Things you need to know about Electronic Viewfinders (EVFs) [Электронный ресурс]. – Режим доступа: https://photographyconcentrate.com/7-things-need-know-electronic-viewfinders-evfs/ (дата обращения: 20.05.2018).
4. Film West Hollywood – сайт для поиска локаций съемок [Электронный ресурс]. – Режим доступа: https://www.filmwesthollywood.com/ (дата обращения: 09.01.2019).
5. Продюсерский центр «Жар Птица» [Электронный ресурс]. – Режим доступа: http://www.zhar-ptica.com/locations/ (дата обращения: 15.01.2019).



Герасимов Александр Игоревич
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41602
Направление подготовки: 09.04.02 – Компьютерная графика и веб-дизайн
e-mail: aligras@ya.ru

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004.01

АНАЛИЗ И ПРОЕКТИРОВАНИЕ МОДЕЛИ ДАННЫХ СОВРЕМЕННЫХ ДИЗАЙН-СИСТЕМ

Герасимов А.И.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

В работе был произведен анализ существующих дизайн-систем крупных компаний, составлен список часто встречающихся атрибутов дизайн-системы, таких как: паттерны, компоненты, цвета, дизайн-токены, утилиты, шаблоны, правила использования и др. сформирован готовый артефакт проектирования дизайн-системы, а именно генерализованная структура дизайн-систем, смоделированная на основе анализа существующих и построенная в нотации UML в качестве диаграммы классов.
Ключевые слова: дизайн, дизайн-система, проектирование, атомарный дизайн, компоненты, шаблоны, токены, дизайн-токены, стили, брендбук.

На сегодняшний день дизайн в продуктовых решениях влияет не только на эстетическое восприятие системы, но и на ее конкурентоспособность на рынке: все больше и больше компаний, разрабатывающих программные продукты, стараются сделать свои решения более привлекательными для потребителя с помощью грамотно построенного дизайна системы и взаимодействия системы с пользователем [1]. Например, компании в Великобритании тратят на дизайн около 92 млн фунтов, что составляет около 2,1% их оборота [2]. В частности при разработке веб-сайтов существуют метрики, определяющие количественную успешность дизайна, которая влияет непосредственно на продажи [3–8]. Например, CRO (Conversion rate optimization – оптимизация коэффициента конверсии). В процессе работы с этим показателем определяется отношение посетителей, которые превращаются в клиентов компании ко всему количеству посетителей сайта за определенное время.

По словам UK Design Council, из 500 избранных акционерных компаний американского фондового рынка (так называемый «S&P 500»), для компаний, которые инвестируют в качественный дизайн, наблюдается увеличение финансовых показателей на 299%, тогда как компании, которые не инвестируют в дизайн или инвестируют в некачественный, финансы растут на 75% [2].

В работах зарубежных (Christopher Alexander, Brad Frost, Marcin Treder, Michah Godbolt, Donella H. Meadows и Skliarenko N.V.) и российских (Алла Холматова) [9–16] исследователей

выделены основные методологии процесса дизайна с использованием системного или компонентного подхода. Однако в настоящее время отсутствуют научные исследования, которые бы описывали полный цикл разработки дизайн-системы, а также содержали в себе анализ существующих. В исследованиях представлены решения проблемы определения только стилистических аспектов разработки дизайн-системы. Однако исследований на тему структурных аспектов дизайн-систем в данной области не имеются.

1. United States Web Design System. Это «Дизайн-система Соединенных Штатов в сфере веб-технологий» предоставляет дизайн-правила и код для того, чтобы государственные министерства США могли быстро и надежно построить сервисы в едином стилевом решении.

Дизайн-система разбивается на несколько главных «блоков»:

– документация;
– компоненты;
– дизайн-токены;
– утилиты;
– шаблоны страниц.

Каждый из разделов отвечает за свои функции, и в сумме все разделы создают полноценное описание дизайн-системы.

Основные функции каждого из разделов перечислены ниже.

Документация. Раздел с документацией состоит из следующих элементов:

– «Вводная» для разработчиков или дизайнеров, которые только начинают работать с дизайн-системой;
– соответствие дизайн-системы требованиям о доступности;
– обучающие статьи и видео;
– примеры использования.

Компоненты. В этом разделе перечислены основные элементы и правила их использования на веб-страницах. Для каждого элемента представлены возможные CSS-классы, состояния и другие необходимые параметры. Примеры готовых компонент: Buttons, Grids, Search bar, Slide navigation и др.

2. Дизайн-токены. Раздел с дизайн-токенами содержит готовые значения для стилей, которые можно использовать на компонентах. Эти значения разделены по тематикам: от цветовых палитр, до прозрачности и теней. К примеру, следующим образом выглядят правила для использования теней на элементах (таблица, где представлено соответствие «дизайн токен–пример–значение»):

Утилиты. В разделе с утилитами хранятся готовые стили, которые можно использовать на компонентах. Эти стили задаются CSS-правилами, к примеру, утилита .bg-primary-darker при компиляции будет превращена в Saas-стиль background-color: color('primary-darker').

К примеру, таблица с соответствием утилиты, «навешиваемой» на текст и значением, которое получается в итоге рендера: шаблоны страниц. Шаблоны страниц содержат в себе несколько примеров веб-страниц, таких как посадочная страница и страница с описанием документации.

Для каждой из страниц указаны правила использования: разделы «Когда использовать», «Когда лучше избегать использование» и «Основные правила», а также список с используемыми на странице компонентами.

Пока что шаблонов страниц всего двое, однако, разработчики обещают в скором времени увеличить это число.

3. Дизайн-система Альфа-банка. Альфа-банк – крупнейший частный банк в России [17]. При разработке дизайн-системы банк преследует следующие цели:

– экономия времени, сил разработчиков и денег банка за счет переиспользования проверенных решений;

– проверка продуктовых гипотез. Из готовых компонентов команды быстро соберут прототип и протестируют его вместе с клиентами;
– единый пользовательский опыт во всех приложениях банка;
– база знаний разработки. В одном месте собран свод правил и лучших практик, помогающих делать качественные и консистентные цифровые продукты. Экономия времени на адаптацию новых сотрудников;
– единая стилистика Альфа-Банка на разных платформах [18].

Дизайн-система разделена на несколько крупных подразделов: Библиотеку компонентов, Принципы, Паттерны, Стиль и Как использовать.

Библиотека компонентов. Самый обширный из представленных разделов. Включает в себя готовые дизайн-компоненты, а именно: пример, код, свойства, методы, а также правила использования. К примеру, компонент «Календарь» имеет пример отображения, код, различные состояния (даты доступны-недоступны, календарь с выбором периода дат, календарь с предвыбранным значением и т.д.). Более того, он имеет готовые параметры, изменяя которые можно изменять отображение календаря (рис. 1). И, в конце концов, правила использования этого компонента описывают правила и требования для использования этого компонента. К примеру: «[Компонент календаря] не используется для выбора дат в далеком прошлом или будущем».

Рис. 1. Компонент «Календарь» в состоянии выбора дат из ограниченного интервала

Принципы. В принципах построения дизайна с помощью дизайн-системы Альфа-банк указывает информацию о методах использования компонент, правила использования единого UX-подхода, манифестирует «mobile-first» подход к разработке дизайна и учитывает различия в iOS и Andoid парадигмах: «в контекстных меню в iOS используются свайпы, а в Android – лонгтапы» [18].

Паттерны. Раздел с паттернами представляет собой так называемый подход «User Story» («пользовательские истории»), которые помогают понять паттерны действий пользователей с системой. К примеру, паттерн «Ввод паролей из СМС и пуш-уведомлений» или правила по верстке и отображению сумм.

Стиль. Раздел со стилем подразделяется на подразделы: «Цвета», «Типографика», «Иконки», «Логотип». Каждый из этих разделов подробно объясняет наборы актуальных палитр, семейств шрифтов и т.д., а также предлагает библиотеку иконок с возможностью поиска по ним, например: «Как использовать». Раздел «Как использовать» описывает нефункциональные требования к системе, в которой будет реализована дизайн-библиотека Альфа-банка. К примеру, в ней указаны требования к технологическому стеку (PostCSS, React, JS ES6+ и т.д.).

Здесь же указаны требования к браузерам.

4. Дизайн-система Рамблера. Дизайн-система Рамблера представлена в виде комбинирования элементов брендбука и правил построения интерфейса. По сути, она представлена в виде

компонент и «всего остального» – информации об использовании логотипа, шрифтов, иконок, паттерна и остальных элементов брендбука.

Компания явным образом разделяет дизайн-систему для разработчика и дизайнера: нажав на кнопку «Дизайнеру», откроется брендбук. Если кликнуть на «Разработчику», откроется Github, в котором будут лежать исходные файлы системы.

Компоненты также имеют свойства, примеры использования, однако, для дизайн-системы нет советов по использованию тех или иных компонент, также как и дизайн-токенов или других важных элементов.

5. Дизайн-система компании «Atlassian». Компания «Atlassian» – лидер на рынке систем управления проектами.

Дизайн-система компании Атлассиан подразделяется на следующие части: Бренд, Маркетинг, Продукт. Каждая функция дизайн-системы четко разграничена в рамках этих частей. К примеру, в части Бренда указаны правила использования цветовой палитры компании, а также правила использования логотипа.

В разделе «Продукта» можно посмотреть на компоненты и правила их использования. Однако раздел с кодом компонент и их свойства лежат в ином разделе, который называется «Пакеты».

Подытожив, можно сказать, что у компании Атлассиан четко разделена информация для бренд-менеджеров, дизайнеров и разработчиков.

Выводы из анализа. Проанализировав несколько дизайн-систем крупных компаний и скомбинировав их элементы, можно сделать выводы, что такого рода системы должны включать в себя следующие элементы:

– компоненты: Код; Примеры использования; Правила использования; Состояния;
– готовые стили для компонент: Паттерны; Дизайн-токены; Утилиты;
– руководство по использованию стиля компании, а именно: Использование иконок; Логотип и его использование; Типографика; Цвета;
– принципы построения дизайна сервисов компании; Детализация дизайна для следующих платформ: Приложения для смартфонов; Десктопные приложения; Веб-приложения;
– документация по принципам и правилам применения дизайн-системы;
– шаблоны и правила их использования.

Рекомендации по построению дизайн-систем на основе выводов из анализа. Как любая система, дизайн-система имеет модель данных. По модели данных проще всего понимать внутреннюю логическую структуру систем [19]. Модель данных дизайн-системы (рис. 2) будет служить практическим применением проведенного анализа в рамках работы и поможет описать правила построения таких систем «с нуля». По сути, представленная ниже модель позволяет моделировать дизайн-систему.

Методы и свойства компонента указаны для React-JS-фреймворка.

В работе была рассмотрена актуальность выбранной темы на основе избранных источников. В процессе исследования были найдены и проанализированы существующие дизайн-системы, их структура и содержание. Были рассмотрены дизайн-системы таких компаний, как «Atlassian», Mail.ru Group, Альфа-банк, Рамблер, а также отдельно проанализирована «Дизайн-система Соединенных Штатов в сфере веб-технологий». В рамках анализа были найдены следующие элементы: компоненты, паттерны, дизайн-токены, утилиты, шаблоны, документация, правила использования и др.

На основе анализа существующих дизайн-систем были сформулированы выводы из анализа, в рамках которого была составлена скомбинированная структура на основе представленных ранее систем [20–22].

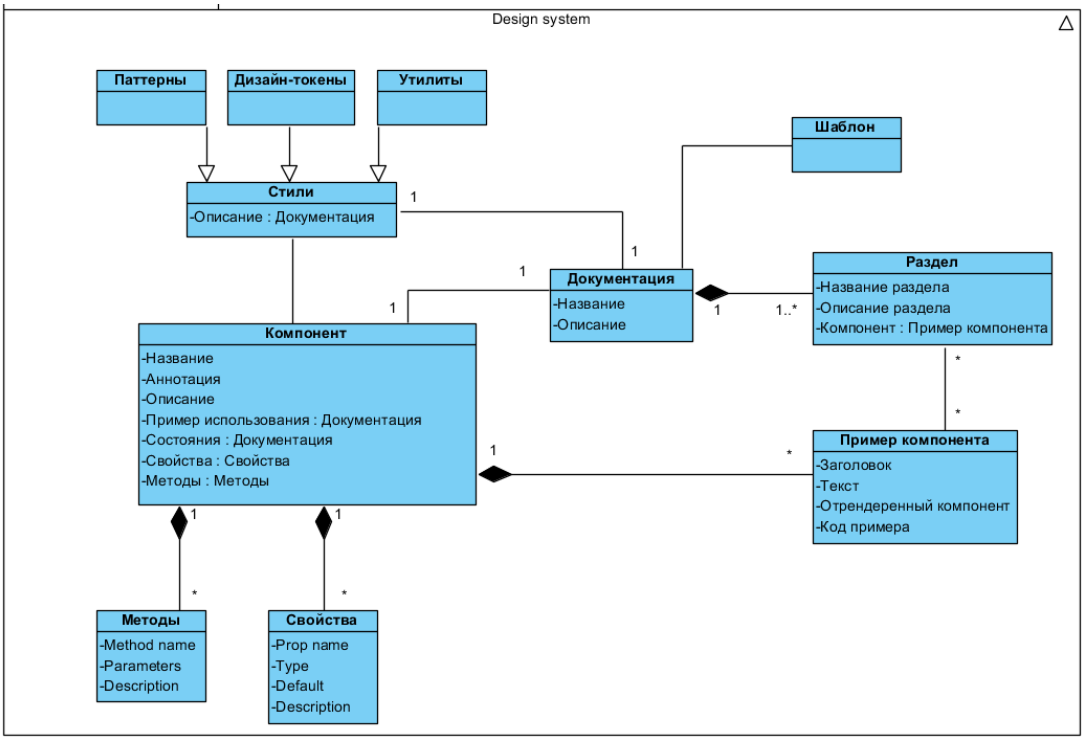


Рис. 2. Модель дизайн-системы на основе анализа аналогов

В рамках практических результатов в разделе «Рекомендации по построению дизайн-систем на основе выводов из анализа» был смоделирован артефакт проектирования дизайн-системы, который описывает ее внутреннюю структуру. Данный артефакт был смоделирован, используя диаграмму классов нотации UML, и представляет из себя готовую структуру, которую может использовать каждый.

Литература

1. Myers B., Hudson S.E., Pausch R. Past, present, and future of user interface software tools // ACM Transactions on Computer-Human Interaction. – 2000. – V. 7. – № 1. – P. 3–28.
2. Livesey F., Moultrie J. Company spending on design: Exploratory survey of UK firms 2008. – University of Cambridge, 2009. – 48 p.
3. Saleh K., Ayat S. Conversion Optimization: The Art and Science of Converting Prospects into Customers, p. 2. – O'Reilly Media, Sebastopol, 2011. – 124 p.
4. Ash T., Page R., Ginty M. Landing Page Optimization. – Wiley & Sons, Indianapolis, 2012. – P. 13–14.
5. Page R. Website Optimization: An Hour a Day, Wiley & Sons, Indianapolis. – Sybex, 2012. – 360 p.
6. Conversion Rate Optimization Report 2014. – London: Econsultancy, 2014. – Retrieved 2.
7. Dvir N., Gagni R. When Less Is More: Empirical Study of the Relation Between Consumer Behavior and Information Provision on Commercial Landing Pages // Informing Science: The International Journal of an Emerging Transdiscipline. – 2018. – V. 21. – P. 019–039.
8. UK Design Council [Электронный ресурс]. – Режим доступа: https://www.designcouncil.org.uk (дата обращения: 14.01.2019).
9. Zweife H.A. Human Sigma Optimization: Engaging Employees and Customers. – University of St. Thomas, Minnesota, 2010. – 162 p.
10. Christopher A., Ishikawa S., Jacobson M. A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series). – Oxford University Press, 1977. – 1171 p.
11. Frost B. Atomic Design [Электронный ресурс]. – Режим доступа: http://atomicdesign.bradfrost.com/ (дата обращения: 14.01.2019).

12. Скляренко Н.В. Парадокс как основа самоорганизации дизайн-систем [Электронный ресурс]. – Режим доступа: https://elibrary.ru/item.asp?id=26644207 (дата обращения: 10.12.2018).
13. Treder M. The Actionable Guide to Starting Your Design System. – UXPing, 2017. – 42 p.
14. Kholmatova A. Design Systems. – Smashing Media AG, 2017. – 150 p.
15. Godbolt M. Frontend Architecture for Design Systems. – O'Reilly Media, 2016. – 198 p.
16. Meadows D.H. Thinking in Systems: A Primer, 2008. – 240 p.
17. 100 крупнейших банков – апрель 2010 [Электронный ресурс]. – Режим доступа: http://www.forbes.ru/lichnye-dengi/banki/46731-100-krupneishih-rossiiskih-bankov-aprel-2010 (дата обращения: 14.01.2019).
18. Дизайн Альфа-Банка [Электронный ресурс]. – Режим доступа: https://design.alfabank.ru/principles (дата обращения: 14.01.2019).
19. Codd E.F. Data Models in Database Management. Proc. Workshop in Data Abstraction, Databases, and Conceptual Modelling (Michael L. Brodie and Stephen N. Zilles, eds.), Pingree Park, Colo. (June 1980): ACM SIGART Newsletter № 74 (January 1981); ACM SIGMOD Record 11(2), February 1981; ACM SIGPLAN Notices 16(1), January 1981.
20. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2007. – С. 93–102, 300–309.
21. Интервью с Roy Stanfield, главным интерекшен-дизайнером в Airbnb. – 2016.
22. Миронюк М.Г., Тимофеев И.Н., Ваславский Я.И. Универсальные сравнения с использованием количественных методов анализа (обзор прецедентов) [Электронный ресурс]. – Режим доступа: https://elibrary.ru/item.asp?id=9269925 (дата обращения: 10.12.2018).

Голубева Виктория Андреевна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41601
Направление подготовки: 09.04.02 – Компьютерная графика и веб-дизайн
e-mail: golubeva.v.a.96@gmail.com

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004.51

ИССЛЕДОВАНИЕ ДИЗАЙНА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С УЧЕТОМ ПСИХОЛОГИЧЕСКОГО ВОСПРИЯТИЯ ЦВЕТА И ФОРМЫ

Голубева В.А.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

В работе обобщены и систематизированы существующие знания по теме психологического восприятия цвета и формы с целью использования полученных результатов в дальнейших исследованиях. Кроме того, выявлены общие положения цветового восприятия, цветовых предпочтений и восприятия формы; выделены основные стандарты психологического восприятия цвета, а также основные принципы использования цвета и формы в дизайне интерфейсов.
Ключевые слова: психология цвета, психология формы, интерфейс, мобильные приложения, дизайн интерфейсов.

Многочисленными исследованиями российских и зарубежных авторов установлено прямое влияние цвета на процесс принятия решения о покупке. Результаты сразу нескольких независимых исследований указывают цвет как основную причину, при выборе конкретного товара. Согласно исследованию окружения на поведения покупателя, 92,6% людей заявляют, что визуальный аспект оказывает наибольшее влияние на решение о покупке, намного более явное, чем вкус, запах и др. В другом исследовании о влиянии цвета в маркетинге ученые обнаружили, что 90% суждений о продуктах основаны только на цвете.

Аналогичная ситуация складывается с интерфейсами веб-сайтов и мобильных приложений. Так, 42% покупателей формируют мнение о веб-сайте на основе его дизайна, включая цветовую схему. Более того, 52% покупателей не посетят ресурс снова, если им не понравилось его оформление. Кроме того, по данным исследований нейробиологов, цвет обладает способностью притягивать определенные сегменты людей, менять предпочтения и влиять на поведение пользователей. Все это дает возможности более направленного и эффективного взаимодействия с целевой аудиторией.

Несмотря на явную практическую значимость данной темы в графическом дизайне интерфейсов, она все еще практически не затрагивается отечественными авторами. Большинство существующих российских работ по цветовосприятию, так или иначе, рассматривают вопрос сугубо с точки зрения рекламы и маркетинга. Аспект же

психологического восприятия формы одинаково мало изучен как отечественными, так и зарубежными авторами.

Современные знания позволяют разделить влияние цвета на два аспекта – психологию цвета и символизм цвета. Важно понимать, что, несмотря на то, что две эти категории часто смешивают между собой и используют как синонимы, на деле они имеют под собой совершенно разные явления.

Символизм цвета, а именно, культурные реакции на цвет, вытекает из множества причин, которые могут быть связаны с национальными традициями, религиозными верованиями и так далее. Например, зеленый цвет священен в Исламе, так как является цветом одежды Пророка, в то время как в Англии зеленый считается неудачным цветом, связанным с расстройством и болезнями [1].

Психология цвета, в свою очередь, связана с физиологической реакцией организма человека на цвет. В сетчатке глаза цвет превращаются в электрические импульсы, которые переходят к гипоталамусу, управляющему гормонами и эндокринной системой человека. Опираясь на это знание, можно с уверенностью предположить, что цвет может оказывать влияние на такие аспекты жизнедеятельности как аппетит, метаболизм, температура, режим сна, состояние нервной и репродуктивной систем человека [2].

На предмет одновременного восприятия цвета и формы были проведены исследования как со взрослыми, так и с детьми, учеными в разных областях науки. Основными полученными выводами можно считать следующие:

– в ситуации ограниченного времени, человек, прежде всего, ориентируется на цвет, и только после этого – на форму, следовательно, именно цвет оказывает первичное влияние на сознание. Формы, тем не менее, также несут в себе психоэмоциональное воздействие и способны изменять физиологическое состояние человека;
– цвет и форма могут, как усиливать друг друга, подчеркивая свои изначальные свойства, так и ослаблять или вступать в противоречие друг с другом.

Стоит также отметить, что рассмотрение психологического восприятия цвета и формы корректно только при использовании дифференцированного подхода. Необходимо отдельно рассматривать категории, выделенные на основе таких признаков как пол, возраст, национальность, особенности зрения и так далее. Проведенные на данную тему исследования позволят сделать следующие выводы [3]:

– гендерные различия цветовосприятия научно доказаны и имеют объективный характер. Большинство исследователей сходится во мнении, что цветовое восприятие среди женщин более разнообразно, чем среди мужчин. Цветовые предпочтения полов также различны, однако четкие закономерности выявить не удалось;
– цветовые предпочтения с возрастом остаются относительно неизменны;
– психологическое восприятие цвета может быть связано с культурно-историческим символизмом и иметь свои особенности в зависимости от национальной принадлежности пользователя;
– цветовые предпочтения субъективны, однако могут быть предсказаны с достаточной долей уверенности.

Обращаясь к общим тенденциям в проектировании интерфейсов, можно заметить, что последние исследования все чаще демонстрируют преимущества максимально упрощенного с точки зрения применения форм и цветов дизайна.

В совместном исследовании Гарвардского университета, университета штата Мэриленд и университета Колорадо [4], ученые обнаружили точные, математически подтверждаемые корреляции между эстетической привлекательностью веб-ресурса, его «простотой» и социальным и демографическими статусами респондента. Чем больше цветов и разнообразных форм было использовано в дизайне интерфейса, тем сильнее было негативное впечатление пользователя от его просмотра. Кроме того, чем старше пользователь и чем выше

его уровень образований, тем более простые цветовые решения он предпочитает. Гендерных различий выявлено не было.

В исследовании, проведенном Google [5], также было установлено, что сложность интерфейса побуждает пользователя оценивать его как непривлекательный. Более того, было статистически доказано, что большинство пользователей предпочитают максимально стериотипичныые интерфейсы, и склоны оценивать их как наиболее привлекательные. Исследователи связывают это с когнитивной легкостью шаблонных интерфейсов и эффектом простого воздействия, согласно которому люди склонны отдавать предпочтения явлениям, с которыми они уже знакомы.

Учитывая вышесказанное, попробуем обобщить и систематизировать рассмотренные выше представления о психологии цвета и формы с целью выявления стереотипов. Помимо цветовых теорий, указанных в работе авторов, также будут использованы трактовки, представляющие собой обобщенный пользовательский опыт (таблица).

Таблица. Стереотипы психологического восприятия цвета

	Иттен	Хэллок	Райт	Обобщенный опыт
Красный	Сила, материя	Скорость, храбрость, страх, веселье	Скорость, смелость, сила, агрессия	Энергия, «срочность»
Желтый	Познание	Скорость, дешевизна, веселье	Оптимизм, экстраверсия, беспокойство	Юность, оптимизм
Синий	Вера	Доверие, безопасность, качество, технологии, храбрость	Доверие, эффективность, логика, рассудительность	Надежность, безопасность
Оранжевый	Самосознание	Дешевизна, веселье	Комфорт, безопасность, изобилие, веселье	Агрессия, призыв к действию
Зеленый	Сочувствие	Доверие, безопасность	Гармония, отдых, мир, спокойствие, скука	Здоровье, расслабленность, покой
Фиолетовый	Благочестие	Храбрость, веселье	Роскошь, качество, интроверсия, замкнутость	Спокойствие
Черный	–	Безопасность, качество, технологии, страх	Изысканность, безопасность, эффективность, угроза	Элегантность, роскошь
Белый	–	Доверие	Чистота, простота,	–
Серый	–	Технологии	Нейтралитет, уныние	–
Коричневый	–	Дешевизна	Тепло, приземленность	–

Основными результатами данной работы можно считать следующие:

– выявленные общие положения цветового восприятия, цветовых предпочтений и восприятия формы: цветовосприятие зависит от культурно-исторических и физиологических особенностей человека, восприятие отдельного цвета зависит от его цветового окружения, цвет и форма могут взаимно усиливать или ослаблять влияние друг друга;
– выделенные основные стандарты психологического восприятия цвета: красный ассоциируется с силой и скоростью, желтый – с оптимизмом, синий – с безопасностью, зеленый – с расслаблением;

– выявленные основные принципы использования цвета и формы в дизайне интерфейсов: цвет и форма должны подчеркивать логику работы приложения, сочетания цветов и форм в интерфейсе должны быть максимально простыми и предсказуемыми для пользователя.

Литература

1. Иттен И. Искусство цвета / Пер. с нем. – 2-е изд. – М.: Изд. Д. Аронов, 2001. – 96 с.
2. Wright A. The Colour Affects System of Colour Psychology [Электронный ресурс]. – Режим доступа: http://www.colour-affects.co.uk/research/ (дата обращения: 04.12.2018).
3. Hallock J. Colour assignment [Электронный ресурс]. – Режим доступа: http://www.joehallock.com/edu/COM498/index.html/ (дата обращения: 04.12.2018).
4. Reinecke K. и др. Predicting Users' First Impressions of Website Aesthetics with a Quantification of Perceived Visual Complexity and Colorfulness // Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. – 2013. – P. 2049–2058.
5. Tuch A.N. и др. The role of visual complexity and prototypicality regarding first impression of websites: Working towards understanding aesthetic judgments // International Journal of Human-Computer Studies. – 2012. – V. 70. – P. 794–811.

Демашов Данил Сергеевич
Год рождения: 1996
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р41622
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: d.danil.s.96@gmail.com

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ФРЕЙМВОРКОВ НА ПЛАТФОРМЕ NODE.JS

Демашов Д.С.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

В работе рассмотрена классификация наиболее известных фреймворков, предлагаемая разработчиками и дополненная рейтингом GitHub. Также представлены результаты сравнительного анализа фреймворков с определением их достоинств и недостатков. Приводятся краткие рекомендации разработчикам приложений по выбору фреймворка с учетом характеристик проекта (размер, возможности, задачи масштабируемости и безопасности).
Ключевые слова: библиотека, фреймворк, Node.js.

В настоящее время по причине повышенной автоматизации и информатизации деятельности человека требуется использование широкого спектра инструментов в различных технологических процессах, в том числе при разработке интерфейсов корпоративных приложений. Однако большое разнообразие современных инструментов веб-индустрии зачастую мешает построить простую в поддержке систему и накладывает ограничения в зависимости от выбранных средств.

Платформа Node.js является одним из популярных инструментов для создания серверной стороны веб-приложения и широко используется различными крупными компаниями [1]. Несмотря на это не существует концепции при разработке приложений и выборе фреймворков на данной платформе, как это происходит при выборе других платформ и языков. Вышесказанное определяет проблему верного выбора каркаса веб-приложения на платформе Node.js [2].

Основными задачами работы являются анализ классификаций фреймворков и результатов сравнительного анализа, представленных в научных исследованиях, аналитических обзорах и публикациях разработчиков, а также определение параметров и характеристик каркасов для веб-приложений, которыми оперируют авторы в исследованиях для проведения сравнительного анализа с целью выявления преимуществ и недостатков фреймворков.

Первоначально требуется составить классификацию каркасов, которыми пользуются разработчики, так как особенности структуры фреймворка влияют на многие параметры будущего приложения, в том числе на возможные размеры, сложность и масштабируемость.

В сообществе выделяют следующую классификацию фреймворков:

1. библиотека HTTP-сервера – набор функций и классов, позволяющих на их основе строить веб-приложения [3], именно поэтому такие библиотеки разработчики относят к фреймворкам, хотя технически каркасами они не являются;
2. API-фреймворк – набор функций для построения системы, основанного на доступе по интерфейсу, а также архитектурный фреймворк, который облегчает создание структуры приложения [4];
3. фреймворк HTTP-сервера – включает в себя такие же функции по работе с командами и маршрутами HTTP, как и библиотека HTTP, однако предоставляет каркас небольшого сервера, построенного на таких функция; на основе сервера разработчик может развивать систему, при этом исключая из кода обработку HTTP-команд [4];
4. веб-фреймворк MVC – фреймворк, использующийся для разработки систем на базе паттерна MVC (Модель-Представление-Контроллер), данная архитектура диктует способ построения приложения именно в такой модели [4];
5. полностековый фреймворк – данный вид фреймворков задает каркас не только для создания системы на стороне сервера, но также определяет тип работы с данными и пользователями на клиентской стороне [4].

Требуется отметить, что при первоначальном развитии Node на рынке не было компаний-монополистов, создавших фреймворк, подходящий под любые веб-приложения. Это привело к тому, что независимые разработчики начали создавать собственные библиотеки и фреймворки для создания веб-серверов и систем. Такой ход породил большое количество различных модулей, каркасов и библиотек, доступных широкому кругу программистов. В связи с этим на текущий момент существует большое количество фремворков на платформе Node, пересекающихся по функционалу, либо имеющих специфические отличия, что негативно влияет на процесс создания конечных продуктов, так как разработчикам требуется выбирать из слишком большого разнообразия.

В табл. 1 представлены известные фреймворки для Node с оценками GitHub на момент написания работы, так как это показывает, насколько известным в сообществе разработчиков является каждый из них [2]. Также указаны тип фреймворков и их возможное применение.

Таблица 1. Классификация фреймворков

Название	Тип	Применение	GitHub Stars
Express	Библиотека HTTP-сервера	Простые приложения	42866
Restify	REST HTTP библиотека	Простые REST API	9113
Loopback	API фреймворк	Веб-приложения, API	12480
Koa	Фреймворк HTTP-сервера	Одностраничные веб-приложения	25318
Hapi	Фреймворк HTTP-сервера	Веб-приложения, API	10865
Sails	MVC-фреймворк	Веб-приложения, API	20321
Meteor	Полностековый фреймворк	Веб-приложения	40857

Для того чтобы наиболее полно отразить особенности каждого фреймворка, необходимо описать каждый из представленных выше:

1. Express – один из первых и наиболее известных фреймворков для Node.js; создана полнофункциональная HTTP-библиотека для веб-приложения любой сложности, упакованная в небольшой размер; благодаря минимализму и высокому быстродействию, Express позволил строить на своей основе другие фреймворки;
2. Restify – данный фреймворк присутствует на рынке приблизительно с появления Node.js и имеет схожие с Express структуру и функции. Restify проектировался специально для построения правильных RESTful (Representational State Transfer) сервисов;

3. Loopback – фреймворк с широкой поддержкой автоматической генерации кода, что снимает необходимость в написании повторяющихся частей приложения; позволяет быстро разрабатывать REST API с полноценной поддержкой баз данных;
4. Koa – облегченный фреймворк с минимальным набором функций; данный каркас подходит для проектирования одностраничных приложений, которые зависят от внешних API;
5. Hapi ориентирован на серверы HTTP и маршрутизацию с помощью команд HTTP; данный каркас подходит для облегченных внутренних подсистем, состоящих из большого количества небольших сервисов;
6. Sails.js является фреймворком MVC, которые спроектирован аналогично структуре Rails/Django; имеет собственную ORM и систему шаблонов;
7. Meteor – полностековый фреймворк MVC; диктует максимально строгую структуру приложения, которая не позволяет разработчикам приложений отклоняться от паттерна MVC; поддерживает организацию веб-сервера, создание мобильных приложений и программ для настольных ПК; может быть интегрирован с клиентскими фреймворками.

Несмотря на то, что данные фреймворки признаны наиболее эффективными с точки зрения разработчиков приложений, они не лишены различного рода недостатков, которые сказываются на последующих веб-приложениях и системах. Данные недостатки, как и достоинства фреймворков представлены в табл. 2.

Таблица 2. Достоинства и недостатки фреймворков

Название	Достоинства	Недостатки
Express	Минимальный набор необходимых функций; небольшой размер; гибкость	Нет строгой структуры проекта; неограниченное подключение модулей
Restify	Достоинства Express; высокая безопасность	Недостатки Express; большое количество повторного кода
Loopback	Автоматическая генерация кода для REST-API; встроенная ORM; встроенные функции взаимодействия с мобильными платформами	Нестандартная схема API; нет собственного API плагинов
Koa	Достоинства Express; элегантная форма написания кода; легче Express	Недостатки Express
Hapi	Поддержка распределенной нагрузки; структура проекта зависит от разработчика	Отсутствие правил структуры проектов; большое количество подключаемых модулей
Sails	Быстрая генерация API; собственная ORM; строгая структура проектов; независимость от клиентской части	Использование только Sails маршрутизации
Meteor	Синхронизации данных между сервером и клиентом; интеграция с клиентскими фреймворками; автоматизации написания кода	Большое количество встроенных функций – не подходит для небольших и средних проектов

Из результатов проведенного анализа следует, что разработчикам приложений требуется первоначально определить необходимые возможности и размер проекта, задачи масштабируемости и безопасности, прежде чем выбирать какой-либо фреймворк. Неверное решение может привести к последствиям, ликвидировать которые будет сложно или невозможно. С другой стороны, верно выбранный каркас позволит максимально эффективно и быстро построить веб-приложение.

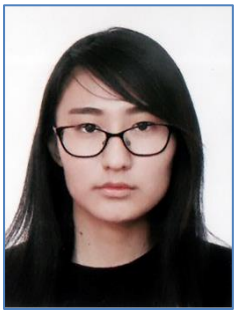
В ходе работы выявлены и проанализированы фреймворки, которые были разработаны для создания приложений на платформе Node.js. Также были определены причины большого

количества каркасов. Из всех фреймворков были определены наиболее популярные согласно рейтингу сообщества программистов и результатов сравнительного анализа, представленного в научных исследованиях. Для общепризнанных каркасов приведены сравнительные характеристики, исходя из научных исследований.

Согласно собранным данным может быть определен фреймворк для приложения по качественным характеристикам. Однако для более детального сравнения каркасов требуется выбор количественных параметров, что является следующим этапом исследования.

Литература

1. Powers S. Learing Node: Moving to the Server Side. – Second edition. – O'Reilly: O'Reilly Media, Inc., 2016. – 304 p.
2. Классификация фреймворков на Node.js [Электронный ресурс]. – Режим доступа: http://nodeframework.com/ (дата обращения: 01.12.2018).
3. Young A., Meck B., Cantelon M. Node.js in Action. – Second edition. – Manning Publications Co., 2017. – 432 p.
4. StrongLoop Node.js Framework Comparison [Электронный ресурс]. – Режим доступа: https://loopback.io/resources/#compare (дата обращения: 02.12.2018).



Дембелова Александра Михайловна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4270
Направление подготовки: 09.04.02 – Мультимедиа-технологии
в искусстве театра, кино и телевидения
e-mail: aliceraliv@yandex.ru

Меженин Александр Владимирович
Год рождения: 1959
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: mejenin@mail.ru

УДК 004.94

МЕТОДЫ ПОВЫШЕНИЯ КАЧЕСТВА 2D-КОМПЬЮТЕРНОЙ АНИМАЦИИ

Дембелова А.М., Меженин А.В.
Научный руководитель – к.т.н., доцент Меженин А.В.

В работе рассмотрены вопросы минимизации отличий двухмерной анимации одушевленных персонажей в конечных анимационных продуктах, созданных с применением технологий 2D-анимации, от традиционной (покадровой) анимации. Предложен алгоритм для улучшения качества анимации в среде Adobe After Effects, позволяющий получать более качественные анимационные продукты.
Ключевые слова: компьютерная анимация, анимация движений, кривые Безье, алгоритм.

Анимационные фильмы в настоящее время широко представлены в рекламных роликах, видеоиграх, мультфильмах и художественных фильмах. Покадровая анимация заключается в прорисовке всех фаз движения, определение в любой момент времени их положения, формы, размеров, цвета. Все кадры при этом являются ключевыми. В компьютерной анимации наиболее востребованной является анимация движений (motion tweening). Анимация движения заключается в рисовании ключевых кадров, соответствующих основным фазам или этапам движения, и последующим автоматическим созданием компьютерной программой промежуточных кадров.

Анимация с максимальным приближением к реалистичным образам в основном заостряет свое внимание на воссоздании реалистичных движений и применении его к детализированным цифровым персонажам. При этом важным является объединение придуманных образов и их действий с реальностью, создание максимального реализма и возможность погружения зрителей в правдоподобную обстановку. Однако часто мультипликационные анимации далеки от реализма и зритель видит неестественные движения персонажей.

Двухмерная 2D-компьютерная анимация (2D vector based animation) одушевленных персонажей создавалась как альтернатива традиционной покадровой анимации для упрощения процесса создания промежуточных фаз движения персонажей в мультфильмах между ключевыми кадрами [1]. В компьютерных программах это может быть получено на основе изменения различных параметров, таких как положение, вращение и масштаб. Компьютер создает промежуточные кадры, которые необходимы для завершения данного

сценария. Таким образом, появляется возможность аниматору экономить значительное количество времени.

Однако разница между анимационными продуктами, созданными с применением этих технологий, остается заметной. Программные системы для создания промежуточной анимации все еще генерируют кадры, которые визуально зрителю представляются отличимыми от кадров, нарисованных человеком. По сравнению с покадровой анимацией 2D-анимация выглядит несколько неестественной. Анимация представляется более естественной, когда она повторяет привычные движения предметов окружающего мира [2]. Можно сделать утверждение, что естественная анимация представляет собой такую анимацию, у которой отличие движений ее объектов от движений объектов естественной среды является минимальным. Под качеством анимации подразумевается степень естественности анимации, т.е. чем более анимация соответствует термину «естественная анимация», тем более она соответствует термину «качественная анимация».

Выдвинута экспериментальная гипотеза, которая проверялась при проведении эксперимента. Она звучит следующим образом: «Анимация, созданная с использованием технологии покадровой анимации, воспринимается зрителями наиболее естественной, чем анимация, созданная с использованием технологии 2D-анимации». Выполнена работа по сбору предварительных данных для определения необходимых действий, которые требуются для проведения основного эксперимента. В ходе сбора предварительных данных выполнен эксперимент. Для более объективного определения естественности анимации подготовлено две пары изображений, которые были предложены респондентам для оценки лучшего восприятия. Каждая пара изображений представляла собой анимацию одного и того же объекта, которая была выполнена с помощью разных технологий: покадровой и 2D- анимации.

Первая пара анимированных изображений представлена на рис. 1. В качестве примера на рис. 1, а, показана последовательность из первых четырех кадров, выполненных посредством технологии 1 – покадровой анимации. На рис. 1, б, – последовательность из первых четырех кадров, выполненных с помощью технологии 2 – 2D-анимации. Вторая пара анимированных изображений аналогично представлена на рис. 1, в, г.

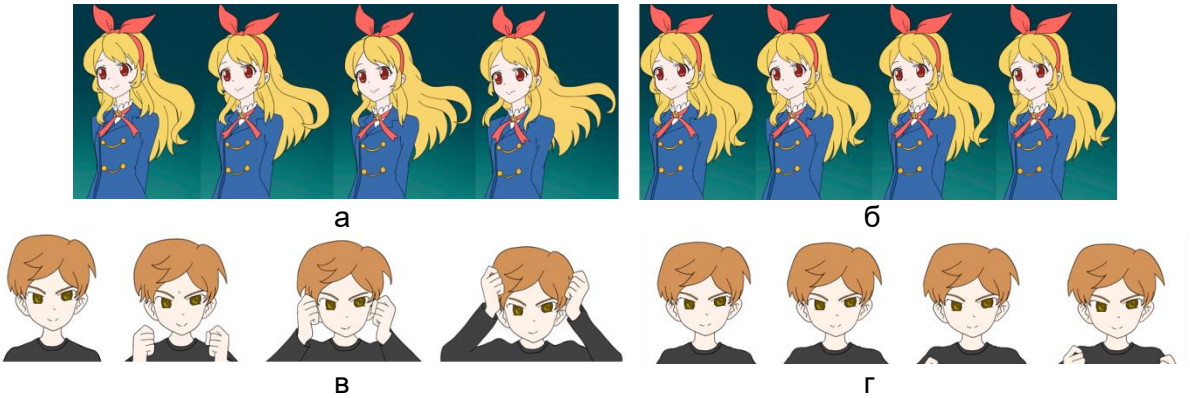


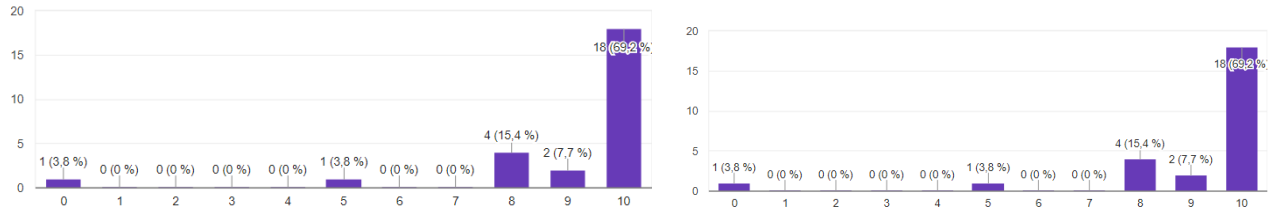
Рис. 1. Последовательность кадров: покадровая анимация (а, в); 2D-анимация (б, г)

Как видно из рис. 1, первые четыре кадра 2D-анимации отличаются от первых четырех кадров для покадровой, так как программное обеспечение автоматически сгенерировало больше кадров за один промежуток времени, чем это было предусмотрено [3] при создании покадровой анимации [4].

Согласно плану исследования, выполнен опрос 26 респондентов, которым предлагалось оценить естественность анимации для каждой пары изображений.

Дана оценка двум вариантам анимации по порядковой балльной шкале от 0 до 10 баллов включительно, где ключевыми баллами являлись: 0 – вариант 1 выглядит совершенно естественнее, 5 – оба варианта выглядят естественно, 10 – вариант 2 выглядит совершенно естественнее.

На рис. 2, а, показан результат опроса для первой пары анимированных изображений, на рис. 2, б, – для второй пары изображений.



а б

Рис. 2. Результаты опроса для первой (а) и второй (б) пары изображений

Согласно схеме независимых испытаний (распределению) Бернулли:

– успех (1) – если по 10-балльной шкале респондент ставит оценку от 6 до 10;
– неудача (0) – если по 10-балльной шкале респондент ставит оценку от 0 до 5.

По результатам опроса для оценки количество возможных вариантов ответов для двух пар анимированных изображений составило 52 ответа. Общее количество оценок от 6 баллов до 10 баллов составило 39. Исходя из полученных данных, частота встречаемости признака в выборке составила 0,75.

На основе полученных данных можно сделать вывод, что с большой вероятностью анимация, созданная с использованием технологии покадровой анимации, воспринимается зрителями наиболее естественной, чем анимация, созданная с использованием технологии твининговой анимации.

Для разработки алгоритма улучшения качества анимации в среде Adobe After Effects предлагается использовать анимационные кривые или графики анимаций. Профессиональный аниматор добивается максимальной естественности за счет графиков анимации. Adobe After Effects – программа для редактирования видео и динамических изображений, а также применения цифровых видеоэффектов. Это программа, которая позволяет монтировать видеоклипы с эффектами, которые сделать в Adobe premiere очень сложно или вообще невозможно. Настройка и редактирование графиков значений и скорости в Adobe After Effects, обеспечивающая естественные и приятные для глаз движения, получаются за счет опыта работы с графиками у аниматора. Предлагаемый метод основывается на редактировании анимационных кривых – графика скорости и графика значений и автоматизирует работу человека, редактирующего графики движений.

Задача предлагаемого алгоритма – создать эффект легкой прерывистости анимации, который не будет препятствовать восприятию ее как плавной и естественной. Суть алгоритма заключается в следующем: при условии, что ключевые кадры уже заданы, создается копия n кадра и копируется перед n+1 ключевым кадром и так до предпоследнего, чтобы получился аналог режима интерполяции hold, когда промежуточные кадры будут полностью отсутствовать. Копия кадра копируется между n и n+1 кадрами. Высчитывается время t между n и n+1 кадрами. Далее берется значение t/2. И копия кадра вставляется в этот промежуток. Тогда первая половина будет интерполирована в режиме hold, а вторая будет двигаться. График скорости следует сделать типа Bezier. Затем отредактировать график скорости так, чтобы его кривая была сильно изломана.

Возможно, задержка анимации за счет первой половины, интерполированной в режиме hold, будет восприниматься зрителем слишком большой. Поэтому в этом случае лучше добавлять копию n-го кадра. Необходимо выяснить, какую часть из промежутка между n и n+1 кадрами нужно взять, чтобы добавить туда ключевой кадр. Измерение степени гладкости кривой диаграммы времени также является нужным для исследования. Предполагается, что при помощи формулы, по которой задана кривая, алгоритм должен изменить ее направление и начертание так, чтобы ее «кривизна» поменялась.

Формальная постановка задачи выглядит следующим образом:

1. входные данные:
   – *frames*[*i*] – массив из ключевых кадров;
   – *num Frames* – количество ключевых кадров в композиции;
   – *intensive* – целочисленное числовое значение, обозначающее степень кривизны Безье [5]. От 1 до 5;
2. промежуточные данные:
   – *new Frames*[*i*] – массив из копий ключевых кадров;
   – *t* – переменная, хранящая в себе время между кадрами;
   – *new Frame Num* – номер установленного ключевого кадра;
3. функции:
   – *get Time*(*frame*1, *frame*2) – возвращает время между двумя ключевыми кадрами;
   – *set Frame*(*time*) – устанавливает ключевой кадр из массива new Frames в заданное время, возвращает номер ключевого кадра;
   – *edit Bezier*(*frame*1, *frame*2, *int*) – меняет параметры кривой Безье между двумя заданными ключевыми кадрами с заданной интенсивностью;
4. выходные данные:
   – результат функции *edit Bezier*, представленный графически.

Предлагаемый метод решения состоит в том, чтобы поочередно из массива *frames*[] выбирать два кадра. Рассчитывается время *t* между *frames*[*i*] и *frames*[*i*+1] кадрами. И копия *frames*[*i*] – *new Frames*[*new Frame Num*] вставляется в этот промежуток при помощи функции *set Frame*(), которая вернет значение *new Frame Num*. Далее задается интенсивность и вызывается функция *edit Bezier*(), которая отредактирует кривую Безье с заданной степенью кривизны.

На рис. 3 представлена блок-схема для данного алгоритма.

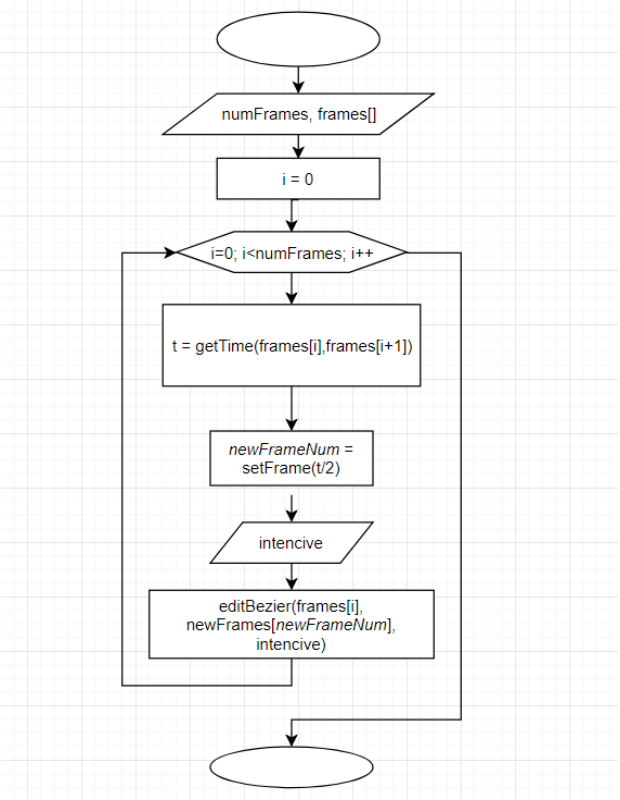
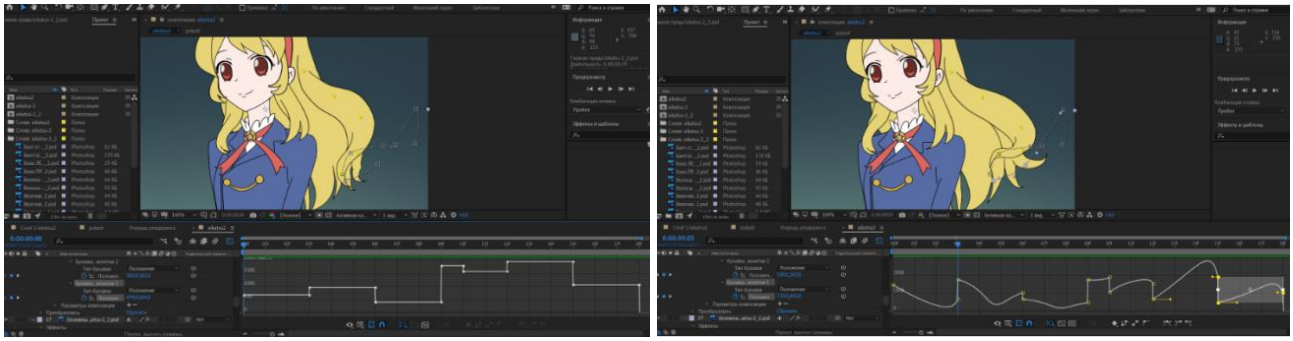


Рис. 3. Блок-схема для предлагаемого алгоритма

Ход алгоритма был поэтапно проделан в среде Adobe After Effects. На рис. 4 представлены значения графика скорости для первого анимируемого изображения до и после применения алгоритма [6–8].



а б

Рис. 4. Внешний вид графика до применения (а) и после (б) применения алгоритма

Таким образом, применение предложенного алгоритма в среде Adobe After Effects позволит значительно улучшить качество анимации.

Литература

1. Ди Хаан Д. Анимация и спецэффекты во Flash / Пер. с англ. – Киев: Вильямс, 2014. – 512 с.
2. Hirose K., Higuchi K. Creating facial animation of characters via MoCap data // Journal of Applied Statistics. – 2012. – № 39. – P. 2583–2597.
3. Hunter G.M. Computer animation survey // Comput. Graph. – 1977. – V. 2. – № 4. – P. 225–229.
4. Cavalier S. The world history of animation. – University of California Press, 2011. – 416 p.
5. Роджерс Д., Адамс Дж. Математические основы машинной графики. – М.: Мир, 2001. – 604 с.
6. Krikke J. Computer Graphics Advances the the Art of Anime // IEEE Computer Graphics and Applications. – 2006. – № 3. – P. 14–19.
7. Алехина Г.В., Козлов М.В., Спивакова Н.Я. Flash – анимация [Электронный ресурс]. – Режим доступа: http://www.e-biblio.ru/book/bib/01_informatika/trehmer_model_i_anim/flash.html (дата обращения: 16.01.2018).
8. Лебедько Е.Г. Системы импульсной оптической локации. – СПб.: Лань, 2014. – 368 с.

Деркунская Светлана Александровна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4260
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: svetlana7-127@mail.com

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004

ИССЛЕДОВАНИЕ ТЕМАТИЧЕСКИХ СИТУАЦИЙ ДИНАМИЧЕСКОЙ ВИЗУАЛЬНОЙ АЙДЕНТИКИ

Деркунская С.А., Шуклин Д.А.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

В работе рассмотрено использование визуальной айдентики в современном дизайне логотипов и фирменного стиля, исследованы тематические ситуации динамической визуальной айдентики.
Ключевые слова: динамическая айдентика, визуальный образ, логотип, фирменный стиль, dynamic identity, visual image, logo, corporate identity.

Динамическая айдентика становится широко распространенной ввиду того, что бренды стремятся постоянно адаптироваться к быстро меняющейся окружающей среде, чтобы оставаться актуальными и востребованными среди конкурентов. На сегодняшний день динамические логотипы перешли в полноценную тенденцию визуальной айдентики, использующие все больше методик для их создания.

Система логотипов, состоящая из множества движущихся частей и контекстуальных стилей, не подходит для каждого проекта, и этот подход требует подробного рассмотрения связи бренда.

Исследование тематических ситуаций динамической визуальной айдентики позволяет выделить преимущества и проблемы, которые могут возникнуть при использовании этой гибкой системы.

Один из таких примеров – логотип г. Мельбурна (ориг. City of Melbourne), который в 2009 году выпустил новый динамический ребрендинг идентичности. Динамический логотип был создан, чтобы стать иконой для Мельбурна, захватив статус города в качестве ведущей столицы с международной репутацией передового опыта, инноваций и лидерства. Он был разработан, чтобы представить организацию как единое целое, позволяя также использовать его для различных суббрендов. Согласно Мельбурнскому веб-сайту, динамичная идентичность в будущем удовлетворяла всем критериям. Динамическая идентичность может использоваться по-разному для разных проектов и на различных платформах, однако, по-прежнему, быть признанной городом Мельбурном. Идентичность использует гибкую систему дизайна, которая может быть интерпретирована на любом источнике информации, что позволит соединить все их проекты и программы с городом Мельбурн (рис. 1) [1].

Рис. 1. Старая и новая версия логотипа «City of Melbourne»

Существует опасность, что разработанный логотип прослужит только до тех пор, пока динамичная идентичность является трендом. Новый логотип является интересным знаком, однако, он не был включен или встроен в элемент наследия или ориентира города Мельбурна. Новый символ города может быть удобно использован в качестве динамичной идентичности для Милана, Манчестера, Москвы или Мемфиса, поскольку у него нет ощущения местоположения или отличительной особенности города.

Основная проблема, с которой сталкиваются при использовании динамической идентичности для брендинга города, заключается в том, что трудности возникают до того, как динамическая идентичность представлена общественности. Проблема развивается вокруг изоляции старшего поколения; динамическая идентичность нацелена на молодое поколение, которое разбирается в цифровых технологиях.

Динамические идентичности, как правило, работают лучше для крупных организаций или компаний, ориентированных на технологии.

Разработанная система идентификации для португальского музыкального зала Case de Musica, расположенного в городе Порту, использует форму здания (рис. 2) в качестве сосуда, представляющего собой динамическую и разнообразную систему идентификации, которая фиксирует различные варианты использования и определения здания и его событий [2].

Рис. 2. Динамический логотип Casa da Musica (2007)

Создание динамической системы идентификации должно объединять различные аспекты организации в одно концептуальное целое, объединять различные части в законченный и отдельный объект. Форма здания и контекст, в котором оно расположено, сообщили окончательный визуально и контекстуальный синтаксис динамической системы идентификации. Рассматривая систему, используемую для изменения аспектов идентичности, контекстная природа идентичности исследуется в ответ на спроектированную среду, в которой существует здание. Эта идентичность и динамическая система, которая определяет ее вариации, иллюстрирует и выделяет сеть отношений, в которую эта система встроена.

После моделирования здания в виде трехмерной формы на компьютере, оно было преобразовано в шесть вариантов идентичности. Каждый из этих логотипов выступает в качестве плоской модели Casa da Musica, и в сочетании с круглой и геометрической

гарнитурой шрифта, фирменный стиль Casa da Musica представляет собой аккуратный и актуальный пример современного дизайна. Допуская шесть различных модуляций идентичности, визуальная гибкость уже включена в систему идентичности. Это учитывает динамическую систему идентичности, которая придерживается культовой формы и названия здания – Casa da Musica (Дом музыки).

Идентификация Casa da Musica – это система, которая осознает свое местоположение в динамичной сети взаимоотношений и надеется выделить эти отношения, функционируя как сосуд, в котором могут быть осязаемы различные внешние эффекты. Создав инструмент, который визуально воспроизводит здание, была разработана идентичность, которая способна подражать многим уровням смысла одновременно. Эти уровни выходят за рамки типичных и статических инструментов, используемых традиционными средствами. Так, с помощью традиционных средств, идентичность способна объединить более широкий уровень целей, не смешивая визуальные и символические компоненты дизайна. С помощью предварительных средств представление здания Casa da Musica улучшает возможности этого инструмента дизайна. Он проходит разные масштабы, от более непосредственного пространства печатного плаката, до людей, которые занимают и формируют здание и вовне к взаимодействию с широкой общественностью. Если бы все эти компоненты не взаимодействовали с торговой маркой, у нее не было бы таких же отличительных характеристик, которые делали бы ее настолько интересной.

В 2011 году Media Lab запустила новую графическую идентичность (рис. 3), чтобы оживить существующий логотип, простую, но прочную цветную полосу, которая была единственной выделенной графической идентификацией учреждения с момента его основания в 1985 году.

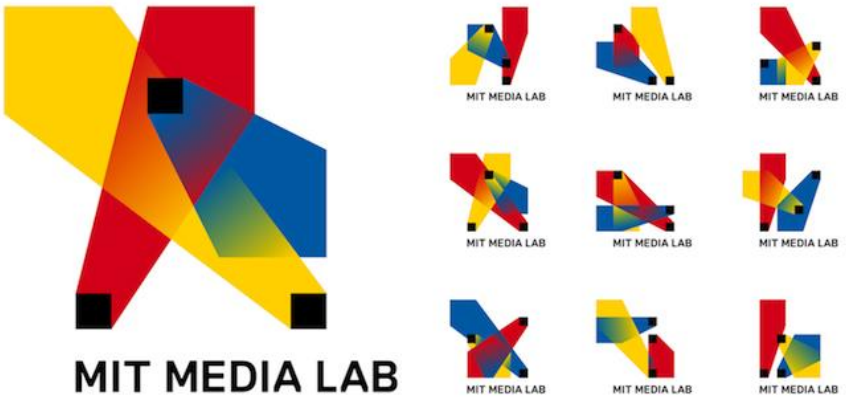


Рис. 3. Динамический логотип MIT Media Lab (2011)

Используя индивидуальный алгоритм, новый дизайн включает три пересекающихся цветных прожектора, которые могут быть организованы в 40 000 различных форм и 12 цветовых сочетаний. Новая система идентификации предоставляет Медиа-лаборатории в течение 25 лет индивидуальные визитные карточки, которые стали отправной точкой для ребренда. Каждый профессор и студент Медиа-лаборатории может выбрать из предлагаемого ассортимента и приобрести свою собственную форму логотипа и цветовую композицию [3].

MIT Media Lab является генеративной динамической системой. Эта порождающая система открывает, по крайней мере, один из элементов динамической системы, которая позволяет идентичности существовать как живой персонаж. Это также позволяет влиять на идентичность внешними данными и идентифицировать MIT в среде реального времени.

В трех тематических исследованиях были освещены некоторые важные аспекты интеграции динамической системы в эти три бренда. Первое исследование основывалось на том, что города фактически не нуждаются ни в каком типе идентичности, ни статическом, ни динамическом. Причина этого в том, что город на самом деле является идентичностью; здания,

люди и культура составляют городскую идентичность. Внедрение динамической системы в качестве идентичности города вызывает лишь недоумение у общественности.

В оставшихся двух тематических исследованиях MIT и Casa da Musica, гибкая система проектирования была успешной, так как в основе этих двух организаций была динамичная среда. И Casa da Musica, и MIT интегрировали свое здание/окружающую среду в свои динамические системы. Элементы, лежащие в основе этих динамических систем, были логическим объяснением развития этой системы проектирования. Это случай функции над формой, тогда как Мельбурн был случаем формы над функцией. Объяснение использования динамической системы для Casa da Music применено из идей, используемых в архитектурном дискурсе к одному из графического дизайна брендинга, и в этом случае динамической системы [4, 5].

Использование динамической айдентики при разработке визуальной системы позволяет выявить сущность представляемого бренда, сделать его уникальным и выделить среди конкурентов. Однако если бренд не связан с динамической идентичностью, т.е. вероятность его потери в динамической системе, в которой он создан.

Литература

1. Finn K. Open Manifesto ‘Identity’. – Melbourne: Finn Creative PTY LTD, 2009.
2. [Электронный ресурс]. – Режим доступа: https://nevolution.typepad.com/theories/2011/03/casa-de-musica-contingent-solutions.html/ (дата обращения: 04.01.2019).
3. [Электронный ресурс]. – Режим доступа: http://www.underconsideration.com/speakup/archives/004431.html/ (дата обращения: 04.01.2019).
4. Van Nes I., Hughea P., van Drunen C. Dynamic Identities: How to Create a Living Brand. – BIS Publishers, 2012. – 48 p.
5. Трофимов Я. Система визуальной идентификации: альтернатива фирменного стиля, правила пользования: Айдентика. – М.: Grey Matter, 2014. – 512 с.

Довгань Александра Николаевна
Год рождения: 1993
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: aleksandra-dv@yandex.ru

Сизиков Валерий Сергеевич
Год рождения: 1939
Университет ИТМО, факультет программной инженерии и компьютерной техники, д.т.н., профессор
e-mail: sizikov2000@mail.ru

УДК 004.932, 519.642.3

РЕКОНСТРУКЦИЯ ИЗОБРАЖЕНИЙ, СМАЗАННЫХ РАВНОМЕРНО И НЕРАВНОМЕРНО

Довгань А.Н.
Научный руководитель – д.т.н., профессор Сизиков В.С.

Работа выполнена в рамках темы НИР № 619296 «Разработка методов создания и внедрения киберфизических систем».

В работе сравниваются два варианта прямой и обратной задач о смазывании изображения по прямолинейной траектории: 1) равномерное смазывание, одинаковое во всех точках изображения (смаз Δ=const). Этот вариант описывается набором 1-мерных интегральных уравнений (ИУ) Фредгольма I рода типа свертки с направлением оси x вдоль траектории смаза, а y – перпендикулярно смазу, а также одним 2-мерным ИУ типа свертки, причем ось x направлена горизонтально, а y – вертикально вниз. ИУ решаются методом регуляризации Тихонова (поскольку задача их решения некорректна) и преобразования Фурье. 2) Неравномерное смазывание изображения нескольких подвижных объектов (смаз $\Delta=\Delta(x)$). Данный вариант описывается ИУ общего типа и решается методом регуляризации Тихонова и квадратур (в случае набора 1-мерных ИУ или кубатур (в случае одного 2-мерного ИУ. Показано, что при неравномерном смазе использование набора 1-мерных ИУ является предпочтительнее 2-мерного ИУ. Получены результаты численных экспериментов.
Ключевые слова: прямолинейное смазывание изображения, равномерный и неравномерный смазы, интегральные уравнения, метод регуляризации Тихонова, MATLAB.

Введение. Рассмотрим одну из актуальных задач обработки искаженных изображений – устранение смазывания изображений путем их математической обработки [1, 2]. Смазывание может быть обусловлено сдвигом устройства регистрации изображения – УРИ (цифрового фотоаппарата, видеокамеры, следящего устройства) или движением самого снимаемого объекта (одного или нескольких человек, автомобилей, самолетов) за время экспозиции. Задача математического устранения смаза состоит из двух задач: прямой задачи (моделирование смаза) и обратной задачи (устранение смаза).

К настоящему времени в ряде публикаций подробно рассмотрен вариант прямолинейного равномерного смаза изображения [1, 2], но недостаточно подробно рассмотрен вариант прямолинейного неравномерного смаза [3] и совсем кратко – вариант

произвольного (неравномерного непрямолинейного) смаза, например, методом «слепой» деконволюции [4, С. 192].

Цель работы – сравнительное рассмотрение двух вариантов прямолинейного смазывания изображения – равномерного и неравномерного. Пример: смазанное изображение бегунов на дорожке, бегущих с одинаковыми, а также с разными скоростями, полученное неподвижным УРИ. Заметим, что в работе [3] рассмотрен случай, когда УРИ за время экспозиции двигалось прямолинейно с некоторой (известной) скоростью $v(t)$. В данной работе рассмотрен случай, когда известен смаз $\Delta(x)$ самих объектов.

Сначала напомним известный случай равномерного прямолинейного смаза [1, 2].

Математическое описание равномерного прямолинейного смазывания изображения. Рассмотрим прямую и обратную задачи.

Прямая задача. Это – задача моделирования равномерного прямолинейного смаза. Она сводится к вычислению интеграла [2]:

$$g_y(x) = \frac{1}{\Delta} \int_{x}^{x+\Delta} w_y(\xi)\, d\xi, \qquad (1)$$

где $\Delta = \text{const}$ – величина смаза; оси x и ξ направлены вдоль смаза, а ось y – перпендикулярно смазу (y играет роль параметра); w_y – заданное несмазанное изображение, а g_y – рассчитанное смазанное изображение в каждой y-строке. Для расчета g согласно (1) разработана m-функция smearing.m [2], а также smear.m – вариант smearing.m при угле смаза $\theta = 0$, а в системе MATLAB для моделирования g имеются m-функции fspecial.m и imfilter.m [4].

Обратная задача. Это – задача устранения смаза. Она может решаться в два подхода.

В первом подходе для устранения смаза решается набор 1-мерных интегральных уравнений (ИУ) Фредгольма I рода типа свертки (при каждом значении y) [2, 5, 6]:

$$\int_{-\infty}^{\infty} h(x-\xi) w_y(\xi)\, d\xi = g_y(x), \quad -\infty < x < \infty, \qquad (2)$$

где

$$h(x) = \begin{cases} 1/\Delta, & -\Delta \le x \le 0, \\ 0, & \text{иначе.} \end{cases} \qquad (3)$$

Здесь h – математически ядро ИУ, а физико-технически – функция рассеяния точки (ФРТ), или point spread function (PSF) [1, 2, 6]. В задаче смазывания функция h, как правило, является разностной, или пространственно-инвариантной, что означает, что смаз равномерный и величина смаза Δ одинакова во всех точках изображения.

Задача решения ИУ (2) является некорректной [7], поэтому используем устойчивый метод регуляризации Тихонова (МРТ) с преобразованием Фурье (ПФ) [1, 2, 7]:

$$w_{\alpha y}(\xi) = \frac{1}{2\pi} \int_{-\infty}^{\infty} W_{\alpha y}(\omega) e^{-i\omega\xi}\, d\omega\ , \qquad (4)$$

где

$$W_{\alpha y}(\omega) = \frac{H(-\omega) G_y(\omega)}{|H(\omega)|^2 + \alpha\, \omega^{2p}} \qquad (5)$$

– регуляризованный Фурье-спектр, или ПФ решения; $H(\omega) = F(h(x))$ и $G_y(\omega) = F(g_y(x))$ – Фурье-спектры функций $h(x)$ и $g_y(x)$, где F – символ ПФ; $\alpha > 0$ – параметр регуляризации; $p \ge 0$ – порядок регуляризации (обычно $p = 1$ или 2). Для выбора параметра регуляризации α разработан ряд способов [2, 7]. Расчет восстановленного изображения по формулам (4)–(5) ведется по разработанной m-функции desmearingf.m [2, С. 137, 330].

Во втором подходе для устранения смаза (а также дефокусирования) используется 2-мерное ИУ Фредгольма I рода типа свертки (ср. (2)) [1, 2, 4–6]:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x-\xi, y-\eta)\, w(\xi,\eta)\, d\xi\, d\eta = g(x,y), \quad -\infty < x, y < \infty, \tag{6}$$

причем оси x и ξ направлены горизонтально, а y и η – вертикально вниз. При этом ФРТ h будет отображаться на плоскости (x, y) в виде узкой полоски (рис. 1) [2, С. 112].

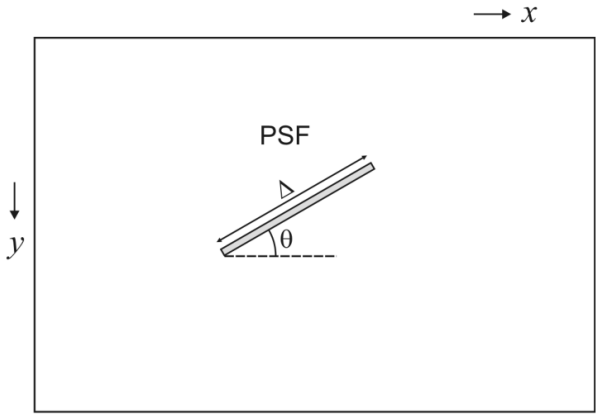


Рис. 1. Функция рассеяния точки (PSF) $h(x, y)$ в виде узкой полоски длиной Δ под углом θ

В этом подходе расчет прямой задачи ведется по m-функциям fspecial.m и imfilter.m [4]. А решение 2-мерного ИУ (6) (обратная задача) методом РТ и 2-мерного ПФ равно $w_\alpha(x,y) = F^{-1}\left(W_\alpha(\omega_1,\omega_2)\right)$, где F^{-1} – обратное преобразование Фурье (ОПФ), или

$$w_\alpha(x,y) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W_\alpha(\omega_1,\omega_2)\, e^{-i(\omega_1 x+\omega_2 y)}\, d\omega_1\, d\omega_2 . \tag{7}$$

В (7) $W_\alpha(\omega_1,\omega_2)$ – регуляризованный спектр (2-мерное ПФ) решения (ср. (5)):

$$W_\alpha(\omega_1,\omega_2) = \frac{H^*(\omega_1,\omega_2)\, G(\omega_1,\omega_2)}{\left|H(\omega_1,\omega_2)\right|^2 + \alpha(\omega_1^2+\omega_2^2)^p}, \tag{8}$$

где $H(\omega_1,\omega_2) = F\left(h(x,y)\right)$, $G(\omega_1,\omega_2) = F\left(g(x,y)\right)$. MATLAB имеет m-функцию deconvreg.m [4] для решения ИУ (6) методом РТ и ПФ согласно (7)–(8).

Мы привели известные формулы (1)–(8) для того, чтобы сравнить различные подходы.

Математическое описание неравномерного прямолинейного смазывания изображения. Учитывая соотношения (1)–(8), рассмотрим неравномерный прямолинейный смаз изображения вдоль траектории смаза. Пусть по смазанному снимку определена некоторым образом зависимость $\Delta = \Delta(x)$ величины смаза Δ от координаты x вдоль смаза.

Прямая задача. В данном случае ФРТ h не будет разностной, или пространственно-инвариантной и прямая задача запишется в виде (ср. (1)):

$$g_y(x) = \frac{1}{\Delta(x)}\int_{x}^{x+\Delta(x)} w_y(\xi)\, d\xi . \tag{9}$$

Для расчета g согласно (9) разработана m-функция smear_n.m.

Обратная задача. Обратная задача в рамках первого подхода запишется в виде набора 1-мерных ИУ Фредгольма I рода общего типа (при каждом значении y) [2, С. 125]:

$$Aw_y \equiv \int_a^b h(x,\xi)\, w_y(\xi)\, d\xi = g_y(x), \quad c < x < d , \tag{10}$$

где A – интегральный оператор; $[a,b]$ и $[c,d]$ – пределы ξ и x. ФРТ h запишется в виде:

$$h(x,\xi)=\begin{cases}1/\Delta(x), & x\le\xi\le x+\Delta(x),\\ 0, & \text{иначе.}\end{cases} \qquad (11)$$

Для решения ИУ (10) нельзя применить ПФ, но можно применить метод квадратур, приводящий ИУ (10) к системе линейных алгебраических уравнений (СЛАУ) [2, С. 126]:

$$\mathbf{A}\mathbf{w}_y = g_y, \qquad (12)$$

где $\mathbf{A}$ – матрица, связанная с h (одинаковая для всех y); $\mathbf{w}_y$ – искомый вектор; g_y – правая часть СЛАУ. Устойчивое решение СЛАУ (12) дает метод регуляризации Тихонова [2, С. 126]:

$$(\alpha\mathbf{I}+\mathbf{A}^T A)\,w_{y\alpha} = \mathbf{A}^T g_y, \qquad (13)$$

где $\alpha>0$ – параметр регуляризации; $\mathbf{I}$ – единичная матрица; $\mathbf{A}^T$ – транспонированная матрица, а $w_{y\alpha}$ – регуляризованное решение в y-строке, равное

$$w_{y\alpha} = (\alpha\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T g_y. \qquad (14)$$

Для компьютерной реализации формул (10)–(14) разработана m-функция desmearq_n.m.

Обратная задача в рамках второго подхода может быть записана в виде 2-мерного ИУ Фредгольма I рода общего типа [8] (ср. (6)):

$$\mathbf{A}w \equiv \int_a^b\int_c^d h(x,\xi,y,\eta)\,w(\xi,\eta)\,d\xi\,d\eta = g(x,y), \quad a\le x\le b, \quad c\le y\le d\,. \qquad (15)$$

Уравнение (15) может быть решено методом квадратур (точнее, кубатур) (ср. [7, С. 167]). Согласно этому методу, каждый из интегралов в (15) заменяется конечной суммой на дискретных сетках узлов по x, ξ, y, η и получается СЛАУ с 4-мерной матрицей $\mathbf{A}$ и 2-мерной правой частью g. Чтобы решить такую СЛАУ, нужно 4-мерную матрицу $\mathbf{A}$ преобразовать в 2-мерную, 2-мерную правую часть g преобразовать в 1-мерную, а получающееся 1-мерное решение w преобразовать в 2-мерное. Хотя (успешная) попытка решения 2-мерного ИУ методом кубатур имела место [7, С. 167–169], тем не менее, это громоздкий метод и его применение для восстановления неравномерно смазанного изображения проблематично.

Можно для решения ИУ (15) применить также метод итераций, например, метод итеративной регуляризации Фридмана [7, С. 272], более простой, чем метод кубатур, однако он требует удачного выбора начального приближения, числá итераций и т.д.

В результате следует признать, что наиболее эффективной является методика (10)–(14), основанная на построчной обработке изображения, согласно которой нужно решать 1-мерные ИУ (10) и СЛАУ (12) с 2-мерной матрицей.

Иллюстративный пример. Был решен следующий численный пример. На рис. 2 представлено исходное изображение семи бегунов на дорожке.

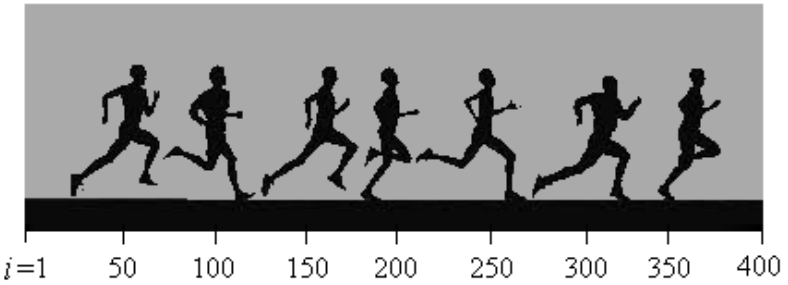
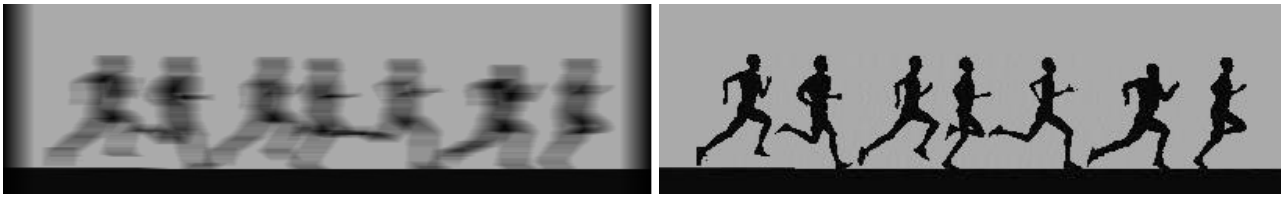


Рис. 2. Бегуны, файл runners.jpg 123×400 пкс (i – номер узла сетки)

Прямая задача. На рис. 3, а, приведен результат выполнения прямой задачи – равномерное горизонтальное смазывание изображения согласно (1) с помощью m-функции smear.m. Величина смаза Δ=20 пкс. Видим, что смазывание изображения весьма значительное.

Смазывание выполнено с размытием краев изображения (опция 'diffusion') для уменьшения эффекта Гиббса – эффекта ложных волн [2, С. 117, 200].

Обратная задача равномерного смазывания изображения. На рис. 3, б, представлен результат решения обратной задачи – построчное восстановление изображения методом квадратур с регуляризацией Тихонова согласно (14) с помощью разработанной m-функции desmearq.m. Параметр регуляризации $\alpha = 10^{-6}$ (выбран путем подбора). Видим, что, несмотря на значительный смаз (рис. 3, а), изображение хорошо восстановилось, причем без эффекта Гиббса благодаря размытию краев изображения (на рис. 3, а).



а б

Рис. 3. Прямая и обратная задачи равномерного смазывания изображения. Смазанное изображение 123×420 (а); восстановленное изображение 123×400 (б)

Обратная задача неравномерного смазывания изображения. Следующий шаг – неравномерное смазывание изображения. Полагаем, что бегуны бегут с разными скоростями v, а значит, у них разные смазы $\Delta = v \cdot T$ за время экспозиции T. На основе рис. 2, авторы определили границы между бегунами и значения смазов Δ бегунов (таблицу). В результате смаз $\Delta(x)$ или $\Delta(i)$ представлен кусочно-постоянной функцией. Каждый бегун в пределах его диапазона имеет свое значение смаза.

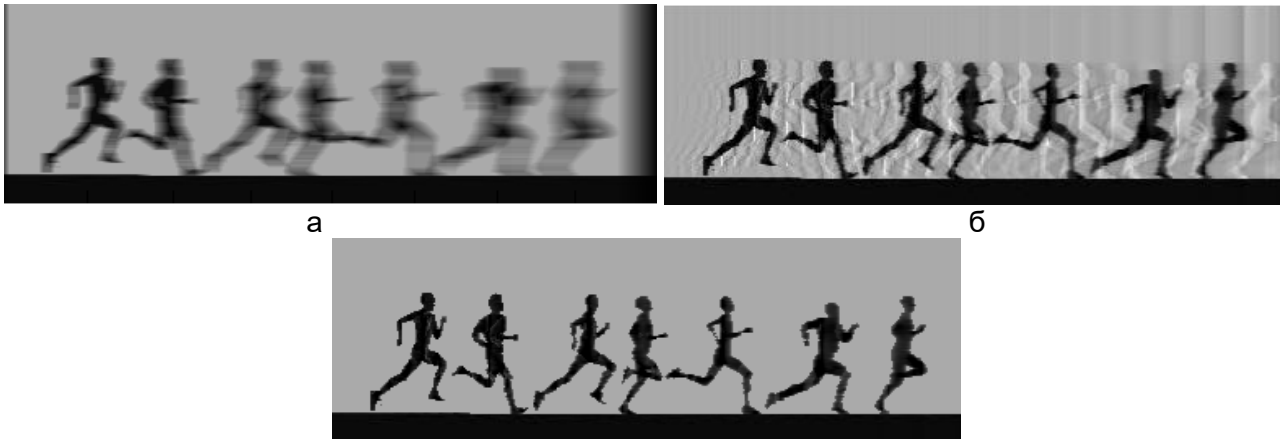
Таблица. Границы между бегунами и значения смазов Δ бегунов

Номер бегуна	1	2	3	4	5	6	7
Диапазон значений i	$1 \le i \le 75$	$75 < i \le 127$	$127 < i \le 182$	$182 < i \le 213$	$213 < i \le 274$	$274 < i \le 339$	$339 < i \le 400$
Значение Δ, пкс	5	8	11	14	17	20	23

На рис. 4, а, представлено изображение, смазанное кусочно-неравномерно, а именно, смазывание нарастает от левого бегуна, у которого смаз равен 5 пкс, к правому бегуну, у которого смаз равен 23 пкс, т.е. смазывание существенно неравномерное. Данное смазывание выполнено согласно (9) с помощью m-функции smear_n.m (с размытием краев).

На рис. 4, б, – результат восстановления изображения (обратная задача) методом квадратур с регуляризацией Тихонова согласно (10)–(14) с помощью m-функции desmearq_n.m. Параметр регуляризации $\alpha = 5 \cdot 10^{-3}$ (выбран путем подбора). Рис. 4, б, показывает, что изображения бегунов восстановились, но с неровным фоном.

Выполнено выравнивание фона: значения интенсивностей на рис. 4, б, больше 100 заменены на 170 (таким выбрано значение фона). На рис. 4, в, – окончательный результат восстановления изображения после выравнивания фона. Видим, что изображения бегунов восстановились достаточно удовлетворительно (ср. с рис. 3, б).



а б

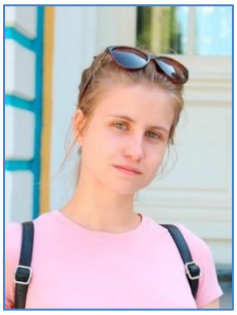
в

Рис. 4. Прямая и обратная задачи неравномерного смазывания изображения. Смазанное изображение 123×405 (а); восстановленное изображение 123×379 (б); восстановленное изображение с выравненным фоном 123×379 (в)

Заключение. Изложена методика восстановления смазанных изображений в случае, когда смазывание является неравномерным (пример: бегуны на дорожке). В этом случае нужно решать или набор 1-мерных интегральных уравнений (первый подход), или одно 2-мерное ИУ (второй подход). В обоих подходах уравнения не являются ИУ типа свертки, поэтому их предложено решать методом квадратур/кубатур с регуляризацией Тихонова. Показано, что первый подход является более предпочтительным, чем второй подход. Численный пример подтвердил это. Методика может быть использована на практике для восстановления групповых изображений нескольких объектов (людей, самолетов, автомобилей), движущихся с разными скоростями, и поэтому за время экспозиции неподвижным УРИ получаем разные смазы Δ на изображении.

Литература

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. – 1072 с.
2. Сизиков В.С. Прямые и обратные задачи восстановления изображений, спектроскопии и томографии с MatLab. – СПб.: Лань, 2017. – 412 с.
3. Тихонов А.Н., Гончарский А.В, Степанов В.В. Обратные задачи обработки фотоизображений // Некорректные задачи естествознания. – М.: Изд-во МГУ, 1987. – С. 185–195.
4. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. – М.: Техносфера, 2006. – 616 с.
5. Сизиков В.С., Экземпляров Р.А. Последовательность операций при фильтрации шумов на искаженных изображениях // Оптический журнал. – 2013. – Т. 80. – № 1. – С. 29–48.
6. Сизиков В.С., Степанов А.В., Меженин А.В., Бурлов Д.И., Экземпляров Р.А. Определение параметров искажений изображений спектральным способом в задаче обработки снимков поверхности Земли, полученных со спутников и самолетов // Оптический журнал. – 2018. – Т. 85. – № 4. – С. 19–27.
7. Верлань А.Ф., Сизиков В.С. Интегральные уравнения: методы, алгоритмы, программы. – Киев: Наук. думка, 1986. – 544 с.
8. Горшков А.В. Улучшение разрешения изображений при обработке данных физического эксперимента и нахождение неизвестной аппаратной функции по программам пакета REIMAGE // Приборы и техника эксперимента. – 1995. – № 2. – С. 68–78.



Дорофеева Мария Игоревна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P4261
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: marie.dorofeeva@yahoo.com

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004.51

ВЛИЯНИЕ КОГНИТИВНЫХ ПРОЦЕССОВ НА ДИЗАЙН ВЕБ-САЙТОВ

Дорофеева М.И.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

В работе рассмотрено влияние восприятия информации на дизайн веб-сайтов, проведен обзор основных законов проектирования пользовательского интерфейса в соответствии с когнитивными процессами. Обоснована необходимость учета опыта пользователя при дизайне интерфейсов и сформулированы главные рекомендации.
Ключевые слова: веб-дизайн, опыт пользователя, когнитивные процессы, пользовательский интерфейс, UX, UI.

Основным фокусом внимания исследователей в сфере UX-исследований (User Experience – опыт пользователя) являются поведенческие сценарии и действия пользователя, связанные с интерактивностью интерфейсов. Это, несомненно, улучшает эргономические и пользовательские показатели цифровых продуктов, однако, не менее значимую роль в формировании опыта взаимодействия играет визуальная коммуникация – графическое представление интерфейса и контента. Таким образом, представление информации также важно, как и управление ею.

Большая часть информации в графическом пользовательском интерфейсе (graphic user interface, GUI), к которым относятся веб-сайты, передается визуальным путем. Следовательно, взаимодействие главным образом строится на визуальном восприятии человеком и осуществляется за счет когнитивных, или так называемых познавательных, процессов в мозгу. К ним относятся: память, внимание, логическое мышление, представление информации, воображение, чувства и способность к принятию решения [1].

Другими словами, процесс взаимодействия можно представить в виде последовательности действий, которые человек совершает постоянно и неосознанно:

1. пользователь изучает визуальную информацию: интерфейс и контент;
2. пользователь анализирует полученную информацию и предполагает, с помощью каких действий он может достигнуть своей цели;
3. пользователь принимает решение о каком-либо действии и совершает его с помощью графических элементов взаимодействия;
4. система реагирует, пользователь вновь считывает информацию. В случае если предположения пользователя не подтверждаются реакцией системы, то имеет место ошибка.

Очевидно, что на качество пользовательского опыта влияют однозначность и ясность представления информации, ее объем и сложность, которые могут вызывать ошибочные действия пользователя. Это, в свою очередь, говорит о том, что чем меньше когнитивная нагрузка (умственная мощность, используемая человеческой памятью) при работе с интерфейсом, тем приятнее и успешнее будет опыт пользователя [2]. А частое несоответствие между ожиданиями пользователя и действиями системы приводит к раздражению, фрустрации и отказу от пользования продуктом.

Дизайн веб-сайтов сводится к вопросу о том, каким организовать и оформить интерфейс, чтобы не затруднять считывание контента, передавать соответствующее сигналы пользователю, вызывать соответствующие эмоции. Каждый элемент на экране является частью графической композиции и имеет свои свойства, которые однозначно определяют его сущность и место в визуальной иерархии [3]. Цвет, форма, размер, схожесть с другими объектами и положение на плоскости экрана – это основные характеристики, играющие роль во взаимодействии с пользователем.

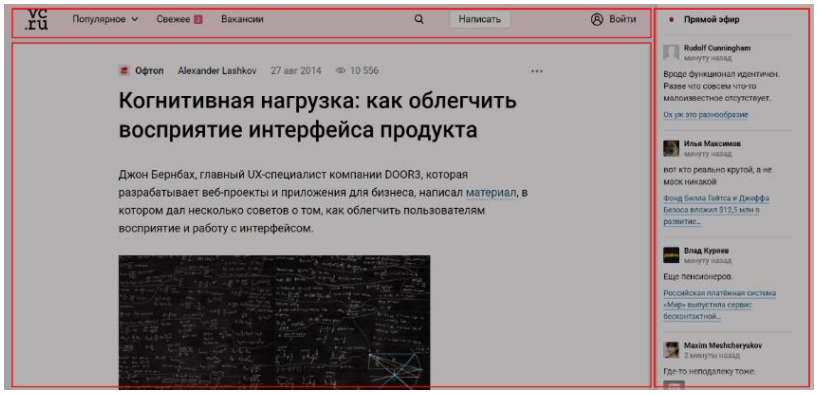
Несмотря на то, что процесс восприятия субъективен, основные закономерности наблюдаются у подавляющего большинства людей, поскольку определяются физиологией, психологией и в некоторых случая культурой и обществом [4]. Основываясь на результатах различных исследований когнитивных процессов, были выделены и сформулированы принципы, которые отражают психологию усредненного пользователя и позволяют с большей уверенностью проектировать дизайн современных веб-интерфейсов (таблица).

Таблица. Основные законы проектирования пользовательского опыта

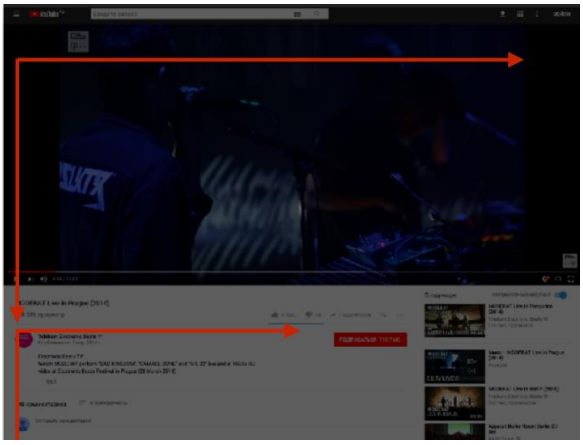
Закон	Формулировка	Применение в дизайне
Закон Хика	Время, необходимое для принятия решения, увеличивается с увеличением количества и сложности выбора	– один из способов облегчить взаимодействие – уменьшить количество вариантов возможных действий; – снижение когнитивной нагрузки, за счет разбивания сложных задач на более простые; – отказ от жестких ограничений действий пользователя
Закон Миллера	Средний человек может хранить около 7 (плюс или минус 2) объектов в рабочей памяти	– осознанное применение в дизайне интерфейсов «числа семь»; – фрагментирование информации путем построения визуальной иерархии – разделение ее на небольшие связанные блоки
Закон фрагментации	Средний человек может хранить около 7 (плюс или минус 2) объектов в рабочей памяти. Человек легче запоминает информацию, если она визуально разделена на блоки	
Закон Якоба, закон пользовательского опыта	Средний пользователь много времени проводит на других сайтах и предпочитает, чтобы любой другой сайт работал так же, как и уже известные ему	– применение в дизайне интерфейсов паттернов пользовательского поведения, чтобы позволить пользователю без затруднений начать работу; – обучение пользователя новому опыту должно происходить постепенно

Закон	Формулировка	Применение в дизайне
Принципы гештальт-психологии: 1. закон близости;	Объекты, которые находятся близко или близко друг к другу, имеют свойство группироваться вместе	Группировать, определять расстояния и отступы между объектами, следует исходя из принципов гештальт-психологии
2. закон сходства;	Человеческий глаз склонен воспринимать подобные элементы в дизайне как целостную картину, форму или группу, даже если эти элементы разделены	
3. закон общей области	Элементы имеют тенденцию восприниматься в группы, если они разделяют область с четко определенной границей	

Среди рассмотренных законов особое внимание следует уделить закону Миллера, так как именно данное заключение стало причиной заблуждения и в течение долгих лет неверно интерпретировалось дизайнерами. Это привело к возникновению «магического числа семь», которое использовалось для оправдания ненужных ограничений в интерфейсах. В действительности Дж. Миллер был сосредоточен не на числе семь, а на концепции фрагментации [5]. Применительно к дизайну, фрагментация означает структурирование информации – разделения ее на небольшие блоки (рисунок, а). Этот принцип повсеместно используется в графическом дизайне и его инструментами являются сетки и макеты.



а



б

Рисунок. Примеры применения закона фрагментации для разделения информации на блоки (а) и известного пользовательского паттерна – F-паттерна в проектировании сайта (б)

Также закон Якоба не взывает к необдуманному использованию шаблонов в веб-дизайне. Главная мысль состоит в том, что человек склонен к построению концептуальных моделей всего, что его окружает. Создавая в уме модель веб-сайта при взаимодействии, пользователь ожидает, что другие сайты будут функционировать и выглядеть в соответствии с этой моделью (рисунок, б). Сильное расхождение с опытом пользователя приводит к когнитивному диссонансу и нередко к невозможности взаимодействия (если пользователь достаточно неопытен или имеет какие-то либо физиологические или психологические ограничения).

Можно сделать вывод, что определяющим фактором хорошего UX является ясность и однозначность представления информации, на основании которой пользователь выстраивает свое поведение. Зачастую пользователь действует по известному ему сценарию (в соответствии с паттернами взаимодействия), поэтому проектировать сайт следует придерживаясь сложившейся и узнаваемой визуальной логике. Немаловажное влияние оказывает группировка элементов, визуальное выделение содержательных блоков и фрагментация, что позволяет снизить когнитивную нагрузку при изучении информации. В свою очередь, снижение когнитивной нагрузки через понятную визуальную иерархию позволяет предупреждать ошибки, ускоряет процесс взаимодействия и вызывает положительные эмоции у пользователя.

Литература

1. Большая Российская энциклопедия [Электронный ресурс]. – Режим доступа: https://bigenc.ru/psychology/text/2077286 (дата обращения: 04.01.2019).
2. Nicholas C. The Shallows: What The Internet Is Doing To Our Brains – W.W. Norton & Company, 2011. – 120 p.
3. Купер А., Рейман Р., Кронин Д., Носсел Кр. Интерфейс. Основы проектирования взаимодействия. – 4-е изд. – СПб.: Питер, 2019. – 460 с.
4. Норман Д.А. Дизайн привычных вещей. – М.: Вильямс, 2006. – 24 с.
5. The Psychology of Design [Электронный ресурс]. – Режим доступа: https://alistapart.com/article/psychology-of-design#section7 (дата обращения: 04.01.2019).

Жмылёв Сергей Александрович
Год рождения: 1992
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: korg@cs.ifmo.ru

Мартынчук Илья Геннадьевич
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4200
Направление подготовки: 09.04.01 – Информатика и вычислительная техника
e-mail: mt4.ilja@gmail.com

Киреев Валерий Юрьевич
Год рождения: 1997
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р3311
Направление подготовки: 09.03.04 – Программная инженерия
e-mail: valeriykireev@gmail.com

УДК 004.75

АРХИТЕКТУРНОЕ ПРОЕКТИРОВАНИЕ СЕРВИСА АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ ОБЛАЧНОЙ ИНФРАСТРУКТУРОЙ

Жмылёв С.А., Мартынчук И.Г., Киреев В.Ю.
Научный руководитель – д.т.н., профессор Алиев Т.И.

Для обеспечения качества разрабатываемых систем и решений в первую очередь необходимо осуществлять архитектурное проектирование будущего решения. В корпоративной и системной архитектурах, в том числе, рекомендуются к проработке следующие ракурсы: мотивация и стратегия, бизнес-процессы, компоненты приложения, технологическая инфраструктура и последовательность их взаимодействия. В работе в полном объеме выполнено архитектурное проектирование, что позволяет сообществам программистов качественно реализовывать сервисы автоматического управления облачной инфраструктурой.
Ключевые слова: корпоративная архитектура, облачные системы, автоматическое масштабирование, системная архитектура, облачные приложения.

Типовое облачное приложение состоит из управляющего модуля, балансировщика нагрузки и горизонтально масштабируемых вычислительных модулей. Узлы, на которых функционируют вычислительные модули, можно включать и выключать в реальном времени, что в рамках крупных центров обработки данных позволяет существенно снижать затраты на электроэнергию. Такой процесс управления вычислительными узлами называется масштабированием облачного приложения.

В работе проанализированы четыре ИТ-решения с открытым исходным кодом (OpenStack, Apache CloudStack, HPE Eucalyptus, OpenNebula), а также по таким косвенным признакам, как интерфейс конфигурации и изменяемые параметры [1], выполнен анализ двух

наиболее распространенных коммерческих ИТ-решений (Amazon AWS и VMware vSphere). В результате анализа сделан вывод, что во всех рассмотренных системах присутствует программный модуль для автоматического масштабирования, в то же время используется один и тот же алгоритм автоматического управления ресурсами на основе граничных значений загрузки оборудования [2]. На рис. 1 представлена обобщенная схема алгоритма управления ресурсами в нотации UML, по которой видно, что не учитываются исторические данные, а также отсутствует какая-либо предиктивная аналитика.

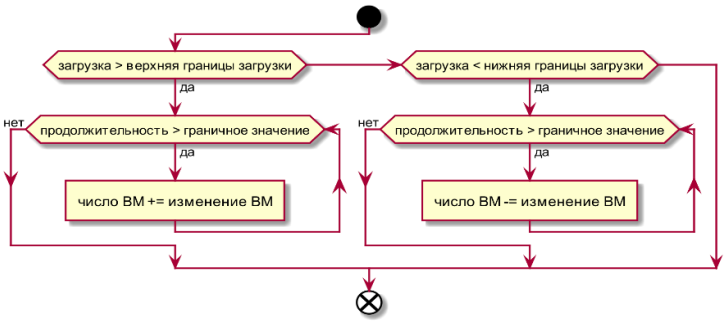


Рис. 1. Обобщенная схема алгоритма управления облачными ресурсами

В открытых источниках отсутствует документация должного уровня, что не позволяет анализировать и качественно реализовывать программные модули для автоматического управления ресурсами.

В соответствии с этим цель работы – обеспечить возможность анализа и реализации ИТ-решений для автоматического управления вычислительными ресурсами облачных приложений. Для достижения цели определены следующие задачи: проанализировать предметную область, выявить требования, возможности, ограничения, сформулировать целевые характеристики; выбрать подход к формированию архитектурных артефактов; подготовить множество диаграмм архитектуры с разных ракурсов для наглядного представления актуального состояния сервисов автоматического управления ресурсами облачных приложений.

В первую очередь выполнен анализ восьми подходов к построению архитектуры, используемых экспертами в более чем 95% случаев: ARIS, TOGAF, Zachman Framework, IDEF, SWEBOK, FEAF, 4+1 view, RM-ODP [3]. Среди рассмотренных способов первые три предлагают более широкий взгляд на корпоративную архитектуру [4]. Поскольку фреймворк ARIS является закрытым корпоративным, его использование сильно ограничено. Модель Захмана использовалась в конце XX века и в настоящее время не учитывает всех аспектов корпоративной архитектуры. Подход TOGAF, непрерывно развиваемый и в настоящее время [5], вобрал в себя достоинства двух предыдущих, поэтому был выбран для данной работы. TOGAF предлагает построить следующие архитектурные представления, покрывающие интересы от владельцев бизнеса до обслуживающего персонала: стратегия и мотивация, бизнес-процессы, ракурс приложения, технологическая инфраструктура, физическое расположение, ракурс уровней.

При анализе предметной области выявлены три заинтересованных стороны, описаны их рычаги управления и сформулировано целевое состояние, позволяющее удовлетворять их потребности. Целевое состояние системы определило ИТ-стратегию, представленную на рис. 2, для которой система должна иметь возможность автоматизированного управления ресурсами с помощью модуля управления ресурсами.

В рассматриваемом предприятии основной заинтересованной стороной является сервис-провайдер [6]. Сервис приложений реализуется облачным приложением, имеющим три компонента: модуль автоматического масштабирования, вычислительный компонент и балансировщик нагрузки. Доступ к сервису осуществляется через веб-интерфейс. Из перечисленных компонентов модуль автоматического масштабирования с позиции ИТ-

стратегии является модулем управления ресурсами и позволяет системе выполнять автоматизированное управление ресурсами.

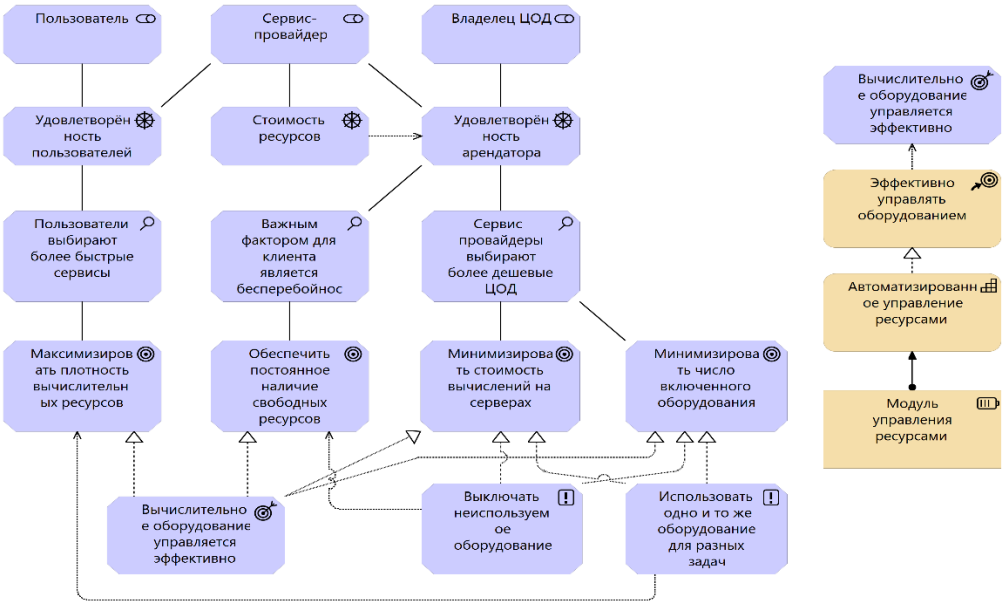


Рис. 2. Диаграммы мотивации и стратегии

Все компоненты должны в качестве платформы использовать технологический сервис, предоставляемый арендуемыми в центре обработки данных серверами. На диаграмме технологической инфраструктуры, представленной на рис. 3, показаны также связи с рабочей станцией оператора и используемые технологии передачи данных.

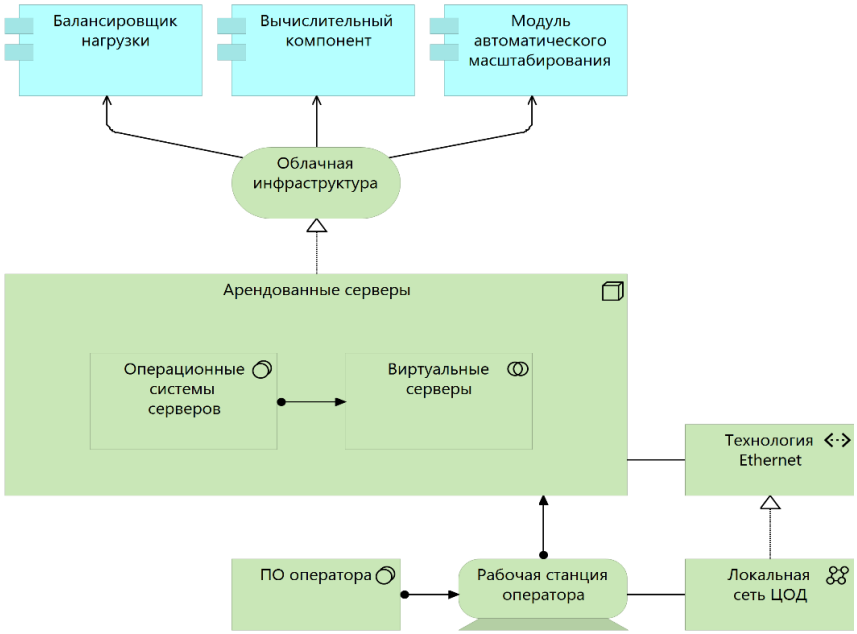


Рис. 3. Технологическая архитектура облачной системы

Ключевые элементы архитектуры и их взаимосвязи представлены в ракурсе уровней, показанным на рис. 4. Достижение целевого состояния обуславливает используемую ИТ-стратегию, в рамках которой системе необходима возможность автоматизированного управления ресурсами. Данная возможность обеспечивается модулем автоматического масштабирования, входящего в состав облачного приложения, предоставляющего сервис приложения, за счет которого исполняется основной бизнес-процесс и пользователям предоставляется внешний бизнес-сервис. Модуль автоматического масштабирования

использует облачную инфраструктуру на базе арендованных серверов из множества серверов центра обработки данных.

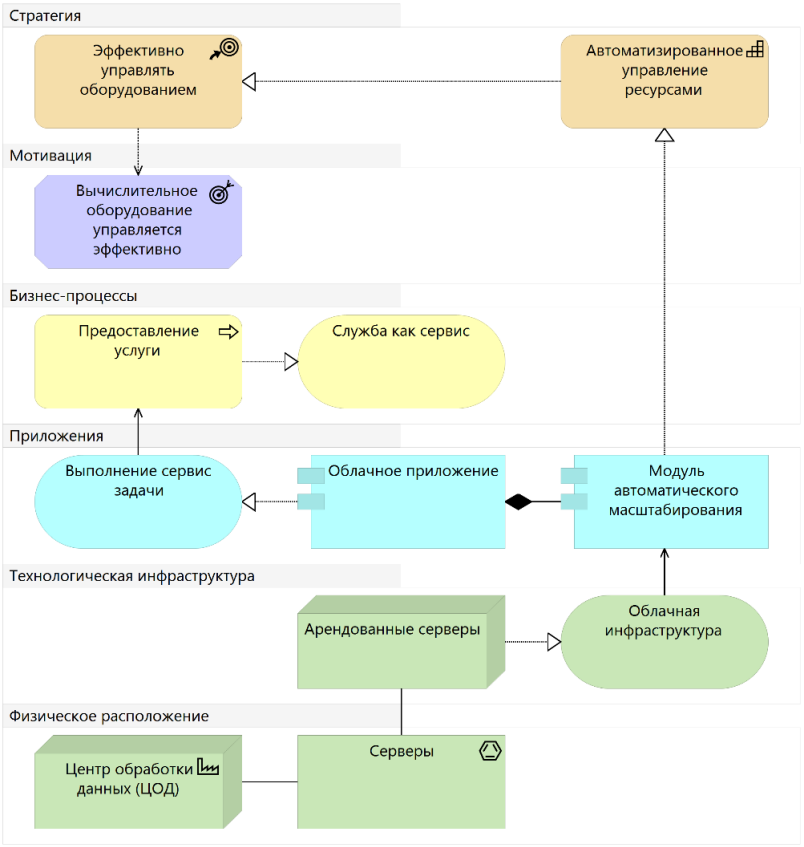


Рис. 4. Ракурс уровней

Проанализирован жизненный цикл модуля автоматического масштабирования и выявлены его основные этапы. Последовательность взаимодействия заинтересованных сторон представлена на рис. 5.

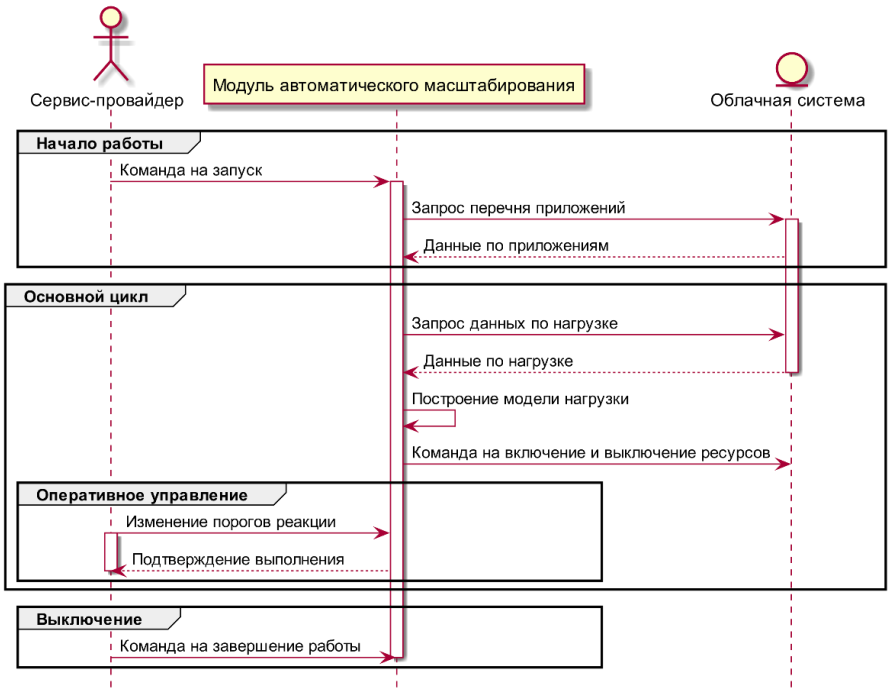


Рис. 5. Диаграмма последовательности взаимодействия

Выделены показанные на рис. 6 компоненты, которые необходимо реализовать в модуле автоматического масштабирования облачной системы. На представленной диаграмме также отражены функциональные блоки, реализующие ключевую функциональность сервиса.

Рис. 6. Диаграмма классов модуля масштабирования

В результате работы разработаны основные архитектурные артефакты: корпоративные – в соответствии с моделью TOGAF, а системные – в нотации UML2. Выполненное проектирование позволяет широкому кругу программистов качественно реализовывать сервисы автоматического управления облачной инфраструктурой, поскольку представленное архитектурное описание содержит совокупность важнейших решений, основывающихся на лучших практиках, используемых в предметной области.

Литература

1. Moreno-Vozmediano R., Montero R.S., Llorente I.M. Iaas cloud´ architecture: From virtualized datacenters to federated cloud infrastructures // Computer. – 2012. – V. 45. – № 12. – P. 65–72.
2. Суммарный трафик участников, передаваемый через MSK-IX // Акционерное общество «Центр взаимодействия компьютерных сетей «МСК-IX» [Электронный ресурс]. – Режим доступа: https://www.msk-ix.ru/traffic/ (дата обращения: 04.01.2019).
3. Rong C.B., Tsai Hsiu-Fen, Chen Chi-Ming. Empirical analysis of server consolidation and desktop virtualization in cloud computing // Mathematical Problems in Engineering. – 2013. – V. 2013. – 11 p.
4. Киреев В.Ю., Мартынчук И.Г., Жмылёв С.А. Аналитические методы исследования нестационарных процессов // Альманах научных работ молодых ученых Университета ИТМО. – 2018. – Т. 2. – С. 258–261.
5. Bogatyrev V.A., Vinokurova M.S. Control and Safety of Operation of Duplicated Computer Systems // Communications in Computer and Information Science, IET. – 2017. – V. 700. – P. 331–342.
6. Жмылёв С.А., Мартынчук И.Г., Киреев В.Ю., Алиев Т.И. Оценка длины периода нестационарных процессов в облачных системах // Изв. вузов. Приборостроение. – 2018. – Т. 61. – № 8. – С. 645–651.

Зиннатулин Фаиль Фидаэлевич
Год рождения: 1994
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4261
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: zinnatulinfail@gmail.com

Шалобаев Евгений Васильевич
Год рождения: 1947
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: shalobaev47@mail.ru

УДК 004.416.6

КОРПОРАТИВНЫЙ ПОРТАЛ КАК ИНФОРМАЦИОННАЯ МОДЕЛЬ ПРЕДПРИЯТИЯ

Зиннатулин Ф.Ф., Шалобаев Е.В.
Научный руководитель – к.т.н., доцент Шалобаев Е.В.

Работа выполнена в рамках темы НИР № 615892 «Исследование и разработка в области информационных технологий».

В работе рассмотрено применение корпоративного портала в качестве информационной модели предприятия; проанализированы отечественные источники информации о корпоративных порталах; рассмотрен корпоративный портал Битрикс24 в качестве информационной модели предприятия, главный функционал Битрикс24.
Ключевые слова: корпоративный портал, информационная модель, интранет, предприятие, сотрудник, веб-сайт, функционал.

Теория информации или теория коммуникации, это математическая теория, сформулированная американским ученым Клодом Шенноном, и признанная разъяснить аспекты и проблемы информации и коммуникации. Хотя теория не является специфической во всех отношениях, она доказывает существование оптимальных схем кодирования, не показывая, как их найти. Например, она преуспевает замечательно в конспектировании требования к инженерству систем коммуникаций и ограничений таких систем. В теории информации термин «информация» употребляется в особом смысле; это мера свободы выбора, с помощью которой выбирается сообщение из множества всех возможных сообщений. Таким образом, информация отличается от смысла, поскольку вполне возможно, что строка бессмысленных слов и значимое предложение эквивалентны по отношению к информационному содержанию [1–6].

В качества места хранения информации можно рассмотреть корпоративный портал, установленный в стенах предприятия и отражающий деятельность этого предприятия.

Корпоративный портал – это корпоративный внутренний веб-сайт (интранет), который предоставляет служебную, общеорганизационную информацию о сотрудниках компании и организации, а также доступ к отдельным, общедоступным веб-сайтам и веб-сайтам вертикального рынка (поставщики и т.д.). Она включает в себя поисковую систему для

внутренних документов, а также возможность настройки страницы портала для различных групп пользователей и отдельных лиц. Множество предприятий внедряют корпоративные порталы в свою информационную инфраструктуру. Корпоративные порталы могут помимо того, чтобы быть заточенными под определенное техническое задание, также предоставлять возможность доработки. Корпоративные порталы обеспечивают возможность хранения информации в своей собственной базе данных (например, MySQL). Все корпоративные порталы можно подразделить на:

- корпоративные порталы для крупных предприятий – численностью свыше 3 000 человек;
- корпоративные порталы для средних предприятий – численностью до 3 000 человек;
- корпоративные порталы для малых предприятий – численностью до 1 000 человек,

а также на:

- коробочные решения – обеспечивает приватность информации о предприятии;
- облачные решения – обеспечивает низкие затраты на начальном этапе, доступность в любой области с доступом в Интернет.

Корпоративный портал был опробован в научно-техническом центре «АО ЗАСЛОН». Предприятие осуществляет разработку:

- проектов программируемых логических интегральных схем;
- низкоуровневое программное обеспечение (ПО);
- ПО под управлением операционной системы (ОС) реального времени;
- ПО под управлением ОС общего специального назначения;
- ПО с сетевыми технологиями;
- ПО с графическим интерфейсом.

Изготовление:

- слесарно-механическая обработка;
- монтаж электронных модулей;
- сборка и монтаж блоков;
- настройка и проведение испытаний.

В качестве корпоративного портала был выбран Битрикс24. Битрикс24 является корпоративным порталом, который предоставляет: единое пространство для всей компании, расписание для встреч, общее дисковое пространство, удобный поиск информации по всему порталу, живую ленту для экономии времени, чат для всего предприятия, онлайн-общение, кастомизацию собственного профиля, планировщика, возможность предварительного просмотра документов и их редактирования через MS Office Online, Google Docs, Libreoffice, Office365.

Битрикс24 как раз является таким порталом, который, может предоставлять, к примеру, такой функционал как:

- хранение структуры подразделений предприятия;
- живая лента новостей;
- управление задачами;
- отчеты к задачам;
- управление группами пользователей;
- сообщения пользователей;
- звонки пользователей;
- хранение сведений о компании;
- хранение документов;
- календарь;
- учет рабочего времени;
- обмен данными с продуктами 1С;
- встроенная панель администратора.

Корпоративный портал хранит совокупность всех сотрудников предприятия, их информацию (местонахождение на территории предприятия, корпоративный мобильный телефон, личный мобильный телефон, должность, ф.и.о. руководителя отдела, интересы, список проектов, в которых участвовал сотрудник), пульс компании.

Пульс компании представляет собой статистику предприятия и состоит из следующих параметров:

- рейтинг – список самых активных сотрудников, которые использовали инструменты корпоративного портала;
- активность – количество действий сотрудников за определенный промежуток времени;
- вовлеченность – процент сотрудников, которые используют хотя бы четыре инструмента.

Пульс формируется с помощью модулей: социальная сеть, лайки, задачи и проекты, чат и видео, диск, CRM. С помощью пульса предприятия можно получать информацию о сотрудниках компании, какие задачи они сейчас выполняют, таким образом, получая информацию о деятельности всего предприятия в целом.

Корпоративный портал Битрикс24 содержит большое количество информации о предприятии и его сотрудниках, поэтому его можно назвать информационной моделью предприятия.

Таким образом, можно сделать вывод, что корпоративный портал является актуальным инструментом для применения на данный момент. Хранение информации и ее качественное, удобное отображение, мониторинг для анализа действий сотрудников предприятия позволяет получать информацию обо всем предприятии в целом, благодаря чему корпоративный портал можно рассматривать как информационную модель предприятия.

Литература

1. Талапов В.В. Основы BIM: введение в информационное моделирование зданий – М.: ДМК-Пресс, 2011. – 392 с.
2. Guruge A. Corporate Portals Empowered with XML and Web Services. – Publisher: Digital Press, 2002. – 313 p.
3. Davydov M.M. Corporate Portals and Ebusiness Integration. – 2001. – 268 p.
4. Битрикс24 [Электронный ресурс]. – Режим доступа: https://www.bitrix24.ru (дата обращения: 27.01.2019).
5. Документация разработчика Bitrix Framework [Электронный ресурс]. – Режим доступа: https://dev.1c-bitrix.ru (дата обращения: 27.01.2019).
6. Основные сведения Bitrix Framework [Электронный ресурс]. – Режим доступа: https://www.1c-bitrix.ru/products/intranet/ (дата обращения: 27.01.2019).

Зиновьев Сергей Сергеевич
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P41702
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: sergey.zinoviev95@gmail.com

Меженин Александр Владимирович
Год рождения: 1959
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: mejenin@mail.ru

УДК 004.92

МЕТОДЫ ОЦЕНКИ КАЧЕСТВА ВИЗУАЛИЗАЦИИ

Зиновьев С.С., Меженин А.В.
Научный руководитель – к.т.н., доцент Меженин А.В.

В работе рассмотрены методы автоматической оценки качества визуализации (рендеринга) 3D-сцен. Для оценки предложено использовать значения гистограммы в отдельных участках изображения.
Ключевые слова: системы реалистичной визуализации, рендеринг, оценка качества визуализации, гистограмма.

Существуют различные подходы для оценки качества визуализации (рендеринга) 3D-моделей и сцен [1–4].

Можно привести один из вариантов: «…качество модели обычно оценивается в зависимости от специфики ее применения:

- компьютерные игры: качество зависит от низкого количества полигонов и правдоподобности;
- анимация – структура скелета, количества суставов и простоты изменения кинематики;
- фотореалистичный рендер – качество определяется реализмом.

Большинство из этих оценок субъективны и не могут быть измерены. Эмпирическое правило:

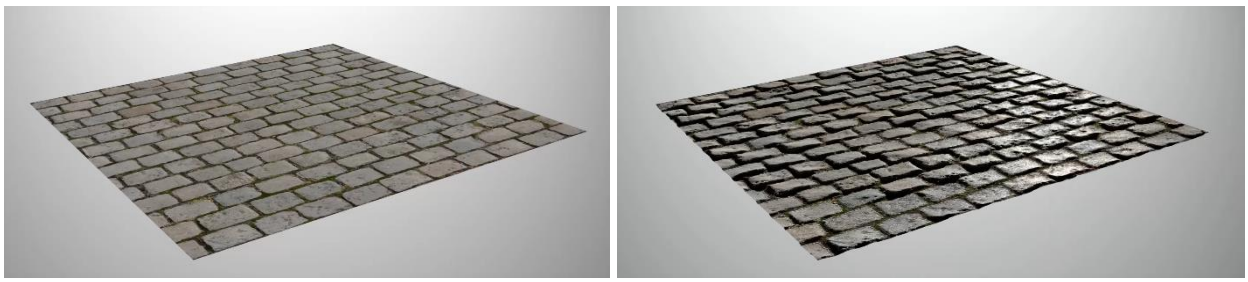
- качество модели – это функция от ее входного размера и реализма, которого она достигает».

Как видно, определенные критерии есть, но измерить их невозможно.

Архитекторы из Орхусской школы архитектуры утверждают, что для разных целевых групп требуется разная репрезентация конечного продукта. Если для инженеров и строителей главное – координаты и габариты, то для обычного пользователя важно увидеть конечный результат. А так как именно пользователи принимают решение о приобретении того или иного продукта, 3D-моделлерам важно создать наиболее реалистичную и приятную глазу визуализацию.

Как вариант, для оценки качества получаемой визуализации можно измерить какую-либо ее составляющую. Таким образом, была поставлена гипотеза: гистограмму изображения, ее значения и характеристики визуализированной 3D-сцены можно использовать для оценки качества визуализации этой сцены.

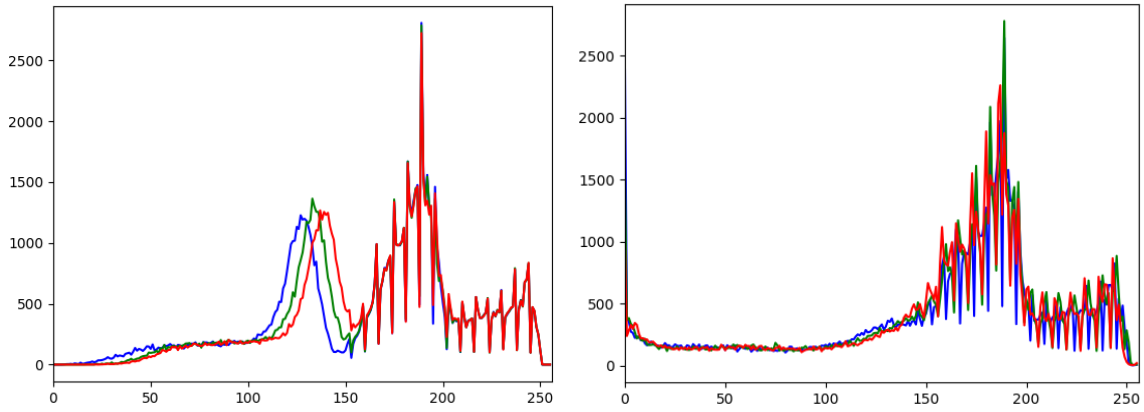
В качестве примера можно привести две визуализации текстуры каменной кладки (рис. 1).



а б

Рис. 1. Визуализации каменной плитки с различными настройками отображения: используется только текстура (а); используются дополнительно карта рельефа и карта отражений (б)

Можно заметить, что изображение на рис. 1, б, кажется более объемным и реалистичным. Сама текстура в данных визуализациях одинакова, разница в картах нормали, смещения и других небольших надстройках.



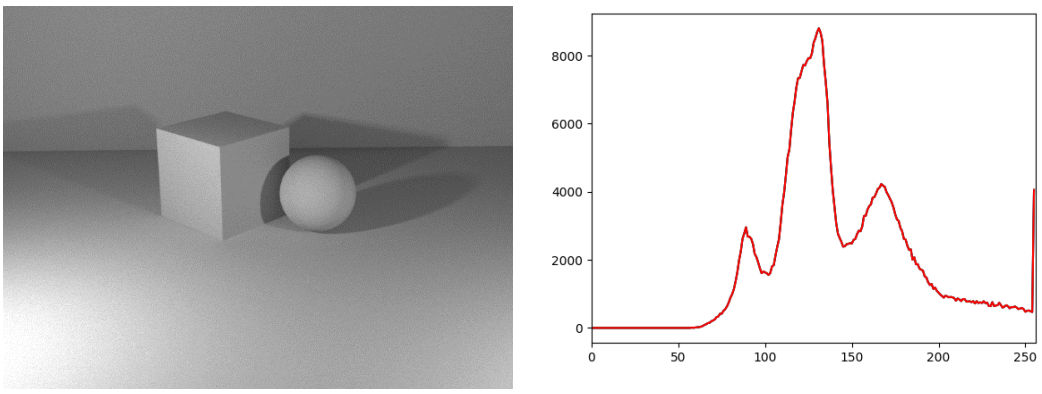
а б

Рис. 2. Гистограммы разных визуализаций каменной плитки (рис. 1)

Гистограммы полученных изображений (рис. 2) довольно сильно отличаются. Но пока что неизвестно, как именно охарактеризовать данную разницу, чтобы о недочетах моделирования можно было сказать только по одному изображению.

Для более детального изучения гистограмм и зависимостей от качества была создана тестовая сцена и получены ее визуализации и гистограммы.

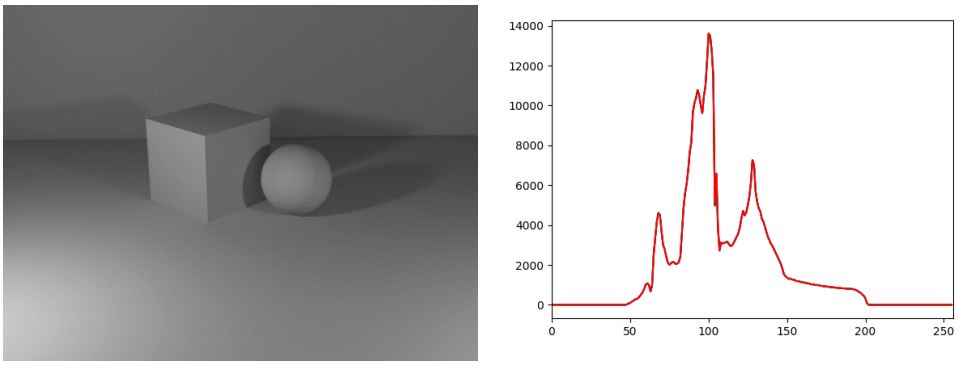
– ART renderer (3D Max) (рис. 3)



а б

Рис. 3. Рендер 3D-сцены с помощью ART renderer (а) и гистограмма полученного изображения (б)

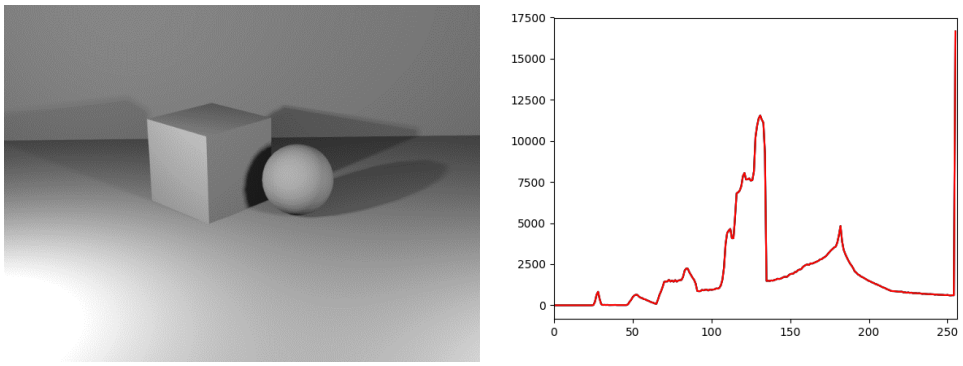
– Corona renderer (3D Max) (рис. 4)



а б

Рис. 4. Рендер 3D-сцены с помощью Corona renderer (а) и гистограмма полученного изображения (б)

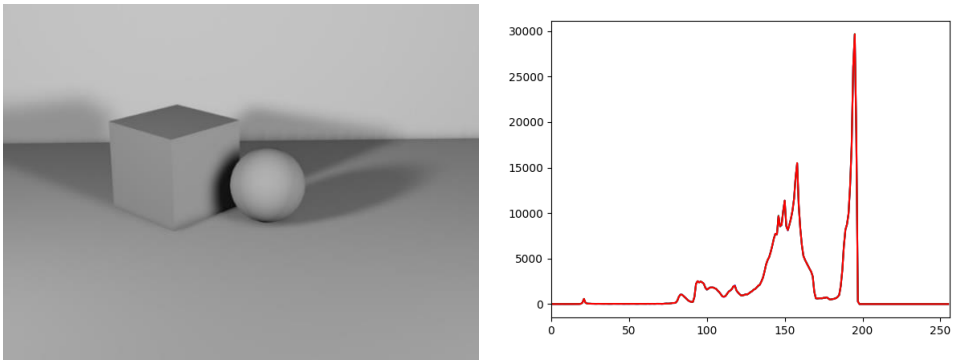
– Scanline renderer (3D Max) (рис. 5)



а б

Рис. 5. Рендер 3D-сцены с помощью Scanline renderer (а) и гистограмма полученного изображения (б)

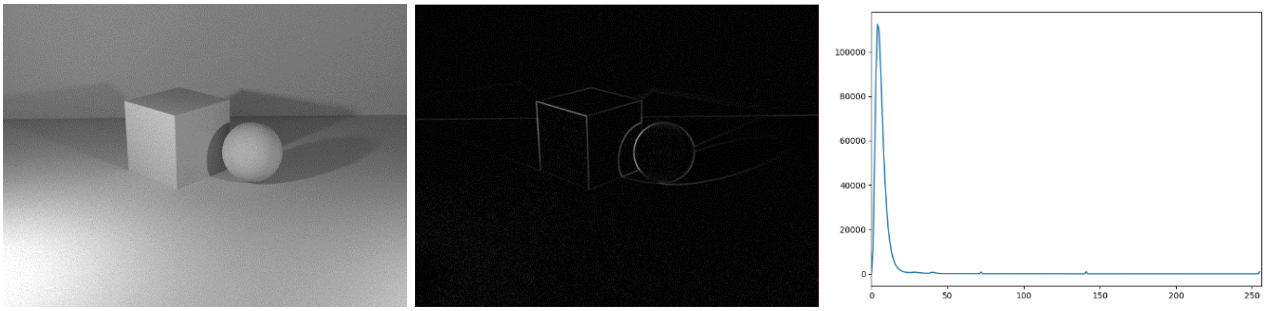
– Team Renderer (Cinema 4D) (рис. 6)



а б

Рис. 6. Рендер 3D-сцены с помощью Team renderer (а) и гистограмма полученного изображения (б)

3D-модели часто заключают в себе не только объект, но и окружение. Чтобы отделить важные сегменты от вторичных, можно использовать разделение на сегменты с помощью гистограммы градиентов (рис. 7). Иначе говоря, блоки, для которых величина градиента невелика и не превышает порог, ограничивающий 75% площади гистограммы, относят к градиентному компоненту. Остальные относят к контурному или факторному компоненту в зависимости от типов соседних блоков.

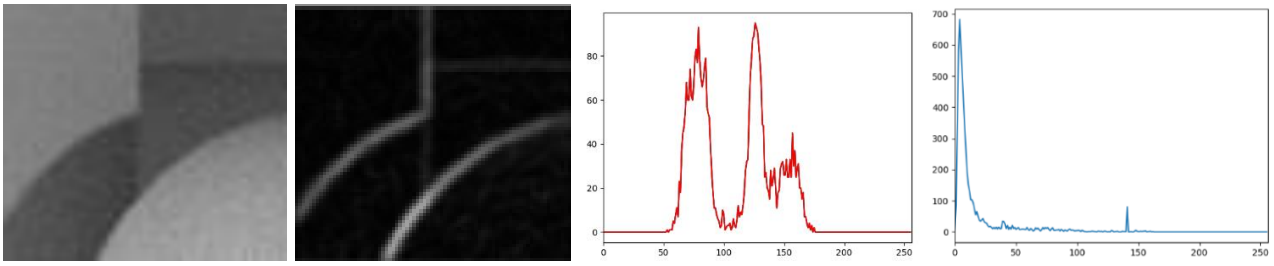


а б в

Рис. 7. Пример гистограммы градиентов визуализации тестовой сцены (ART renderer (3D Max)

Таким образом, на отдельных участках изображения 3D-визуализации можно будет судить о том, какого характера там находится объект: однотонный, контрастный или же это фон.

Разбивая на блоки, получим отдельные изображения, на основе которых будем получать отдельные гистограммы (рис. 8).



а б в г

Рис. 8. Фрагмент визуализации с низкими градиентами (а) и гистограммы исходного фрагмента (б) и градиентов (в, г)

Используя полученные блоки изображения, и измеряя гистограмму на них, получим вес блока (его важность, в зависимости от того, объект это или фон), и распределение яркости в этом переходе.

Получены гистограммы градиентов для сбора данных об однотонности и размытости изображения. Методом разбиения на сегменты осуществляется оценивание изображения не в общем, а по сегментам с объектом сцены. В дальнейшем планируется изменять композицию и характеристики сцены и отслеживать изменение гистограммы для нахождения закономерностей, вывод математической формулы, для нахождения конечного коэффициента качества визуализации. Чтобы добиться объективности оценки различных моделей, планируется провести онлайн-опрос, где респондентам необходимо будет выбрать из пары визуализаций ту, что по их взглядам кажется качественней.

Литература

1. Соболевский А.А. История 3D-графики: от Евклида до наших дней, ВШЭ [Электронный ресурс]. – Режим доступа: https://www.hse.ru/news/communication/150125816.html (дата обращения: 04.01.2019).
2. Красильников Н.Н. Цифровая обработка 2D и 3D изображений. – СПб.: БХВ-Петербург, 2011. – С. 130–131.
3. Южиков В. Blind Deconvolution – автоматическое восстановление смазанных изображений [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/175717/ (дата обращения: 04.01.2019).
4. Satya Mallick – Histogram of Oriented Gradients, 2016 [Электронный ресурс]. – Режим доступа: https://www.learnopencv.com/histogram-of-oriented-gradients/ (дата обращения: 04.01.2019).

Иванова Полина Владимировна
Год рождения: 1994
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4260
Направление подготовки: 09.04.02 – Компьютерная графика и web-дизайн
e-mail: Polina-jdnk@yandex.ru

Готская Ирина Борисовна
Год рождения: 1957
Университет ИТМО, факультет программной инженерии и компьютерной техники, д.педагог.н., профессор
e-mail: iringot@mail.ru

УДК 004.5

АНАЛИЗ ИНТЕРАКТИВНЫХ КОМИКСОВ ДЛЯ ДЕТЕЙ

Иванова П.В.
Научный руководитель – к.педагог.н. Готская И.Б.

В работе проанализированы классические комиксы, выделены общие черты и закономерности, составлен перечень отличительных особенностей комикса как жанра, были проанализированы именно детские комиксы, а также выделены нюансы составления комикса для детей. Были проанализированы зарубежные и отечественные интернет-сайты и приложения с интерактивными комиксами по теме исследования, выявлены методы внедрения интерактивных элементов в комикс и составлен перечень с особенностями интерактивности в данном жанре.
Ключевые слова: интерактивные технологии, воспитание и развитие, дизайн, комиксы для подростков, электронные комиксы, мобильное приложение.

Комикс как жанр обладает собственным набором характерных черт. Ниже представлен перечень этих характеристик:

- нарративность – логичное повествование какой-либо истории от завязки до самого финала, основным инструментом которого выступает раскадровка, имеющая определенную логичную последовательность и сюжет. В комиксах отношения иллюстраций и текста куда сложнее, чем в иллюстрированной книге, где текст занимает главную роль, а иллюстрации лишь подчеркивают его самые выразительные моменты [1];
- языковые средства. Так как текстовая часть значительно меньше, чем в книгах, то она должна обладать большей выразительностью. Потому на помощь приходит обилие языковых средств: гипербола, метафоры, ирония, риторический вопрос, восклицание, умолчание, эпитет, эмоциональное обращение, уличный сленг, жаргонизмы. С помощью данных средств автор управляет темпом и ритмом повествования, представляет читателю характеры и манеры поведения героев, передает настроение;
- действующие лица. Так же, как и в классической литературе, в комиксе есть главный герой, его спутники, антагонист, второстепенные персонажи. Персонажа раскрывают настолько, насколько потребуется, учитывая его роль в сюжете. Это можно делать не только с помощью текста, частоты участия персонажа в диалогах, но также и степенью его прорисовки. Таким образом, можно верно акцентировать внимание читателя на определенных персонажей;

- графические средства выразительности. Форма и цвет хорошо подчеркивает определенный жанр. Агрессивность подчеркнута острыми формами в образах персонажей – например, Росомаха. Французская исследовательница А. Барон-Карве отмечает особую важность цветовой палитры в комиксе. Для придания таинственности следует использовать оттенки серого, зеленого и синего; намекать о приближении чего-либо неприятного следует в теплых цветах, таких как охра или коричневый; оранжевые же и красные тона сопровождают бурную сцену или ссору; пастельные тона говорят о романтике или спокойствии [2];
- жанры. Каждый комикс, как и любое традиционное произведение, имеет определенный жанр или набор жанров. Есть научно-фантастические, исторические, мистические комиксы. Комикс-детектив, комикс-комедия, комикс-экшн, комикс-драма;
- особенности типографики. Шрифты имеют большое значение в комиксе. К ним применяются те же средства выразительности, что и в иллюстрациях: размер, цвет и форма, экспрессивность, толщина линий. Более эмоциональные сцены прорисовываются подробнее, количество текста минимальное, размер шрифта максимальный, много цветов. Нейтральные же сцены могут быть вовсе монохромными, шрифт сдержан и монотонен, чтобы не отвлекать внимание от смысла;
- мультипликационность. Графическая составляющая комикса может быть очень нереалистичной, позволительно изменять анатомию человеческого тела, лица, придумывать новые формы фигур, новых существ и персонажей. Порой образы героев гиперболизированы, даже гротескны, порой схематичны;
- слияние текста и графики. Из традиционной книги можно без потери для смысла извлечь весь иллюстративный материал, что невозможно в комиксе. Изображения задают действия, обстоятельства, а филактер (облако мыслей или слов) показывает взаимодействие персонажей, мысли, общий смысл происходящего. Отсутствие любой составляющей разрушит последовательность истории и смысл;
- вербальные и невербальные компоненты. Вербальные компоненты включают в себя весь буквенный текст комикса. Есть два подвида текстов: авторская речь (заголовки, резюме, титры, комментарии) и реплики/мысли персонажей [3]. Речь персонажей помещена в «облако» – специальное пространство, исходящее от головы персонажа – филактер. К невербальным компонентам относится вся графика, представленная последовательностью рисунков, заключенных в рамки (фреймы) и параграфига, транслирующая фоновую дополнительную информацию (усиливает экспрессивность и эмоциональность комикса, создает анимацию графической части).

В детской литературе должны гармонично сочетаться нравственные, познавательные и эстетические начала. При работе с детскими изданиями необходимо помнить о природе детского образного мышления, и раскрывать содержание при помощи образов, использовать метафорические средства, так как «наивный детский реализм» не разграничивает вымысел и реальность [4]. Ниже представлены нюансы детского комикса:

- гиперболизация. Данная особенность характерна и для классических комиксов, но к детскому комиксу нужно отнестись более тщательно, не должно оставаться двусмысленных посланий, и, чем младше целевая аудитория, тем все посылы должны быть понятнее, структура логичнее, смысловая нагрузка меньше;
- цвета и формы также подчиняются данному требованию. Количество цветов уменьшается, оттенки становятся чище и ближе к основным цветам цветового круга, формы упрощаются, приближаются к стандартным геометрическим фигурам, мелкие детали и второстепенные элементы можно вовсе не использовать;
- количество фреймов на странице уменьшается, линия последовательности чтения упрощается до порой ровного разделения страниц на равные фреймы, для самых маленьких можно также подчеркнуть переход от фрейма к фрейму линиями или стрелками;

– форматы страниц. Из вышеперечисленных особенностей касаемо текстов и иллюстраций вытекает логичное использование больших форматов страницы;
– поля. Для данных форматов характерны широкие поля, увеличенное количество пустого пространства на странице. Иногда большие поля сами могут служить частью графического повествования;
– антропоморфизм и анимизм – наделение растений, животных, неодушевленных предметов человеческими свойствами и восприятие реальности как живой и одушевленной. Животные и предметы живут в человеческой среде, носят одежду, имеют человеческое поведение и эмоции. Но эти особенности не должны сильно влиять на изображенные на иллюстрациях вещи и явления, ведь они должны быть узнаваемы и конкретны;
– шрифт. В него можно вносить игровые моменты, превращать его в иллюстрацию частично или полностью. Важно использовать шрифты крупных кеглей (14–18), которые удобны для медленного чтения [5]. Длинные полосы текста, особенно мелким кеглем, недопустимы в детских изданиях, так как подобные тексты трудны для восприятия.

Правила построения комикса немного меняются, когда в него включается интерактивность. Нужно учитывать особенности представления подобного комикса. Необходимо отметить, что для разных идей можно использовать различные степени внедрения интерактивности, поэтому данные нюансы будут актуальны не для каждого интернет-комикса. Ниже перечислены аспекты, на которые нужно обратить внимание при проектировании интерактивного комикса.

1. Активная камера.
2. Возможность изменить сюжет. Интерактивный комикс может быть и линейным, но добавление этой функции делает комикс ближе к игре и увеличивает целевую аудиторию. В некоторых комиксах учтено множество действий, влияющих на развитие сюжета: реплики, ответы на вопросы других персонажей, совершенные и не совершенные действия. В итоге получается продукт, который можно читать несколько раз и, изменяя свои действия, наблюдать за сменой сценария. Не стоит забывать, что внедрение данной функции требует длительной работы сценариста и увеличивает в разы время и стоимость разработки нескольких линий сюжета.
3. Игровой компонент. Игровой компонент можно внедрить в комикс в разных вариациях. Например, атака осуществляется с помощью множественного нажатия на фрейм в заранее обозначенных местах. Другой подход – головоломки. Можно вставить классические загадки и игровые элементы, а можно сделать игру на основе уникальных особенностей комикса как жанра. Пользователь должен в режиме реального времени так успевать менять местами фрагменты комикса (мини-кадры), чтобы герой комикса мог спастись от преследователей.
4. Действие в ответ на клик. Клик не обязательно должен нести функцию перелистывания страниц или участие в игре. Можно включать в сюжет скрытые бонусы, не обязательные для прохождения игр и никак не влияющие на сюжет. Это может быть легкая анимация фоновых предметов и персонажей.
5. Движения. Есть много способов реализовать движение фона и персонажей. Все очень индивидуально и требует подхода исходя из целей и настроения всего произведения. Ниже перечислены основные направления реализации движений:
   – движение фреймов. В особо опасных моментах, для того чтобы показать волнение, азарт или адреналин, можно использовать резкую, с эффектом потряхивания анимацию всего фрейма, где происходит волнительное действие. Более классический и универсальный подход – плавная анимация нового фрейма, чтобы показать, что появился новый элемент;
   – движение фона. Для того чтобы подчеркнуть спокойствие и безмятежность ситуации, можно использовать большие анимированные фреймы только с фонами. Например, показать кадр с пейзажем, где шевелится трава или ветер качает кончики деревьев.

Данный прием лучше не использовать совместно с анимацией самого персонажа – может вызвать рассеивание внимания и смещение центра;

- взаимодействие фона и персонажа. Для концентрации внимания на то, куда идет и что делает персонаж на фрейме, лучше двигать фон относительно персонажа. Если же требуется показать саму суть ухода, то следует перемещать самого персонажа по фрейму. Для усиления эффекта скорости можно комбинировать анимацию. Основные правила здесь – постоянно следить за тем, чтобы не смещался центр внимания, и чтобы не случился перегруз фрейма и всей страницы.

6. Реакция на положение планшета. Такой вид анимации делает комикс более живым, усиливает вовлечение читателя в повествование.

В результате анализа работ в данной области выявились наименее проработанные на зарубежном и отечественном рынке ниши интерактивного материала для детей, а именно, интерактивный комикс, работа над прототипом и презентацией которого будет проведена в дальнейшем.

Литература

1. Ефлонов В. Комиксы как вид искусства [Электронный ресурс]. – Режим доступа: https://shazoo.ru/2017/04/21/50658/komiksy-kak-vid-iskusstva (дата обращения: 13.11.2018).
2. Baron-Carvais A. La bande dessinée. – Paris, 1985. – 219 p.
3. Козлов Е.В. Комикс как явление лингвокультуры: знак - текст - миф. – Волгоград: Изд-во ВолГУ, 2002. – 183 с.
4. Ситникова Е.В. Иллюстрация как визуальный компонент контента детских изданий // Филологические науки. Вопросы теории и практики. – 2016. – № 2-1(56). – С. 62–68.
5. Волкова В.В., Газанджиев С.Г., Галкин С.И., Ситникова Е.В., Тарбеев А.В. Дизайн периодических изданий: учебное пособие. – М.: Изд-во МГУ им. М.В. Ломоносова, 2014. – 152 с.

Калёнова Ольга Вячеславовна
Год рождения: 1994
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: ovkalyonova@corp.ifmo.ru

Перл Иван Андреевич
Год рождения: 1986
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: ivan.perl@corp.ifmo.ru

УДК 004.04

СРАВНЕНИЕ ПОДХОДОВ К ОБРАБОТКЕ ГЕТЕРОГЕННЫХ ДАННЫХ

Калёнова О.В., Перл И.А.
Научный руководитель – к.т.н. Перл И.А.

В работе приведен обзор современных направлений исследований в области гетерогенных систем и данных. Были получены два противоположных направления: гомогенизация путем формализации и интеграции, и изучение гетерогенных систем и данных умышленно без гомогенизации. Помимо этого, в работе приводится описание технологических подходов к обработке гетерогенных данных с ориентацией на процесс.
Ключевые слова: гетерогенные данные, гетерогенные системы, обработка данных с ориентацией на структуру и природу.

С каждым днем растет объем цифровой информации в мире. Однако информация ничего не значит, если ей не пользоваться. В связи с этим растет роль алгоритмов искусственного интеллекта для обработки информации и получения на их основе новых знаний. Полученные новые знания могут быть применены для разных целей. В их число входят как повседневные задачи людей и бизнеса, так и глобальные задачи человечества и его развития.

Обрабатываемые данные могут быть поделены по разным классификаторам на разные категории. Гетерогенные данные предполагают собой наличие данных, получаемых из разных источников и (или) неоднородных по своей структуре. Противоположным является понятие гомогенных данных, состоящих из однородных данных.

Покажем подробнее понятие гомогенности и гетерогенности на примере распределенных баз данных (БД). Гомогенная распределенная БД состоит из объектов БД одного и того же типа, включенных в однородную (гомогенную) информационную сеть, использующую одинаковые протоколы и интерфейсы доступа ко всем объектам сети, включая локальные БД. В то время как гетерогенная распределенная БД состоит из объектов (локальных БД, модулей, контуров) различных фирм-производителей, включающих разнотипные технические средства и фрагменты разнородных сетей разной топологии, которые могут работать по различным протоколам [1].

Понятие гетерогенных данных связано с понятием мультимодальных данных тем, что источники мультимодальных данных являются гетерогенными и (или) мультимодальные объекты гетерогенны между собой.

К факторам гетерогенности можно отнести следующие [2]:

- различные типы данных (логический, целый, вещественный, объектный и др.);
- различная природа данных (числовые массивы, тексты, медиаданные);
- различные типы (модели) БД – реляционные, иерархические, объектно-ориентированные, сетевые, многомерные и т.п.;
- различные форматы представления данных;
- различия в степени распределенности систем хранения данных;
- различной природы данные (например, тексты, изображения, сигналы);
- различная степень достоверности и точности данных, измеряемых в различных масштабах и единицах измерения;
- различная степень и форма структурированности данных и т.п.

Причины появления гетерогенных данных и объектов также весьма разнообразны и глобальны:

- технологическая (информационная) сингулярность и датапокалипсис;
- каскад бифуркаций;
- законы сложных систем;
- рыночная экономика.

Современные исследования, затрагивающие понятие гетерогенных данных, можно разделить на два больших направления:

1. интеграция данных (Data Fusion) – включает в себя также задачи очистки, унификации и хранения данных в едином формате;
2. управление гетерогенной системой – это направление включает в себя как работы с системами с гетерогенными объектами и субъектами управления, так и с гетерогенными процессами, происходящими в системах.

Таким образом, исследования могут быть отнесены к одному из двух направлений в зависимости от фокуса решаемых задач и выбираемых подходов. Например, исследования из области интернета вещей (IoT) могут относиться как к первому направлению, так и ко второму в зависимости от задач исследователей.

OLTP и OLAP. Существует два наиболее значимых подхода к системам, обеспечивающим доступ к хранилищу данных и используемых в системах анализа данных:

1. OLTP – Online Transaction Processing – основа подхода в том, что данные нужны нужные и вовремя;
2. OLAP – Online Analytical Processing – сосредоточенный подход на обработке больших объемов данных, различных агрегатных функциях для данных, структурированных по многомерному принципу.

Рассмотрим то, на сколько требования к реализации подхода OLTP покрывает требования к системам анализа данных. Как видно из таблицы, OLTP-подход более применим для оперативного доступа к данным, но не обеспечивает возможность анализа данных. В связи с этим OLTP-подход принято использовать для обеспечения доступа операторов к хранилищу данных. В то же время OLAP-подход используется для обеспечения доступа аналитика, где скорость и нагрузка на систему менее важны, по сравнению с необходимостью в гибкости анализа данных и специфичности запросов к хранилищу.

Таблица. Покрытие требованиями к реализации подхода OLTP и системам анализа данных

№	Характеристика	Удовлетворение требованиям OLTP
1	Степень детализации хранимых данных	нет
2	Качество данных	нет
3	Формат хранения данных	нет
4	Допущение избыточных данных	нет
5	Управление данными	да

№	Характеристика	Удовлетворение требованиям OLTP
6	Количество хранимых данных	нет
7	Характер запросов к данным	нет
8	Время обработки обращений к данным	да
9	Характер вычислительной нагрузки на систему	частично
10	Приоритетность характеристик системы	частично

Классический процесс жизненного цикла обрабатываемых гетерогенных данных (рисунок) состоит из трех уровней и механизмов.

1. извлечение данных, их трансформация и загрузка;
2. транспортировка данных – передача данных из/в хранилище;
3. непосредственное хранение данных.

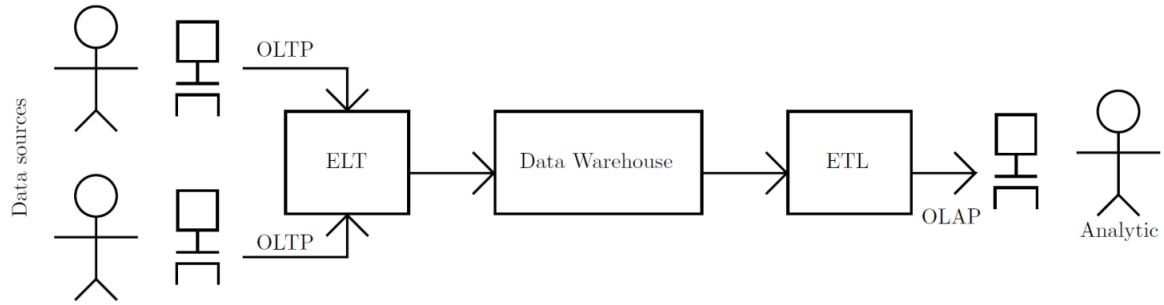


Рисунок. Процесс обработки гетерогенных данных

В соответствии с этим строится процесс, аналогичный структурной схеме на рисунке. Рассмотрим подробнее каждый из уровней механизмов.

Транспортировка данных. Для работы с большими объемами гетерогенных данных, существует два основных механизма: OLTP и OLAP. Исторически первый из них OLTP – сфокусирован на снижении времени пересылки данных и стабилизации при этом нагрузки на систему. Однако OLTP не применим для обработки и анализа данных, так как не удовлетворяет условиям, выдвигаемым к системам анализа данных (таблица): из этого списка OLTP в полной мере, удовлетворяет только пунктам 5 и 8, и частично удовлетворяет условиям в пунктах 9 и 10.

Для извлечения максимальной выгоды из преимуществ OLTP, его используют на этапе загрузки данных в хранилище, что обеспечивает быстрое добавление данных в хранилище.

В то же время OLAP имеет не стабильную нагрузку на систему – она может изменяться от 0% до 100%. Время работы OLAP не гарантируется как минимальное. Однако OLAP позволяет выполнять различные операции по анализу данных и работает чаще всего с агрегатными функциями. В связи с этим OLAP используется на стороне аналитика, когда время роты системы менее важно, по сравнению с результатом.

Хранилище данных. Следующим важным механизмом, требующим внимания при проектировании системы для работы с гетерогенными данными и их анализа, является хранилище данных.

Широко распространенным подходом для хранения больших гетерогенных данных является Data Warehouse. Он обладает 4 основными свойствами:

1. предметно-ориентированность;
2. интеграция данных;
3. не изменяемость данных;
4. хранение истории.

Рассмотрим основные, наиболее важные, понятия хранилища данных с позиции хранения в них гетерогенных данных.

– Таблица фактов и таблица связей. Таблица фактов, как правило, хранит в себе сами данные, которые будут анализироваться, в то время как таблица измерений хранит в себе

описания этих данных. Таблица измерений хранит в себе так же индексы для таблицы фактов. Таким образом, на основе данных таблицы фактов и измерений строятся OLAP-кубы данных и Data Mart, которые используются для ответа запросы аналитиков и пользователей хранилища данных.

Основная проблема в использовании таблиц фактов и измерений для гетерогенных данных в том, что в ней не могут быть данные в разном формате: если есть таблица с ценой товара, то она должна быть в одинаковой валюте.

- Киоски данных. Пользователей хранилища данных можно подразделить на группы в зависимости их области занятости. Для них формируется киоск данных, который является специфической выборкой и представлением данных для уменьшения объема хранилища, к которому группа пользователей делает запрос. Такие витрины данных могут быть иерархически организованы и объединены в собственную топологию. Такой подход призван уменьшить число операций объединения таблиц большого размера и общую нагрузку на физические ресурсы сервера хранения данных.
- Data Vault. Эволюционно следующим подходом к разработке способа хранения интегрированных данных, предназначенных для анализа, является архитектура Data Vault. Сформулированный подход Data Vault ориентирован на существующие бизнес-процессы, в том числе и на процесс разработки программного обеспечения.

Такой архитектурный подход существенно больше подходит для обработки и анализа гетерогенных данных, однако, он все еще требует интеграции данных.

- Data Lake. Data Lake (Озеро данных) не занимается анализом данных, а лишь хранит их до востребования. Часто Data Lake используют для хранения промежуточных или не формализуемых данных. Однако существенной его проблемой является заболачивание – усложнение доступа к нужным данным, если его периодически не очищать.

ETL vs ELT. Базовая парадигма обработки данных строится из трех этапов:

1. извлечение данных из их источников;
2. трансформация. Может включать в себя несколько последовательных этапов преобразования данных;
3. загрузка на целевую платформу.

На основе этого существует два основных шаблона выстраивания процесса обработки данных [3]:

1. Extract – Transform – Load (ETL);
2. Extract – Load – Transform (ELT).

Шаблоны отличаются друг от друга этапами, на которых выполняется преобразование данных и, исходя из этого, оборудование. Выбирая между двумя шаблонами, следует сфокусироваться на следующих вопросах:

1. имеет ли целевая платформа возможность, как база данных, преобразовывать данные после того, как они были на нее загружены? Если нет – скажем, вы загружаете в облачный сервис, а не в базу данных, – тогда требуется преобразовать данные до его загрузки;
2. имеет ли инструмент интеграции возможность преобразовывать данные? Если нет, то может быть не целесообразно использовать его для преобразования данных налету.

Таким образом, в работе произведен обзор современных направлений по работе с гетерогенными данными на основе открытых источников научных публикаций. В результате было выделено два основных направления. Первое направление исследований связано с интеграцией, нормализацией и формализацией (гомогенизацией) гетерогенных данных. Второе направление занимается изучением гетерогенных данных и систем, умышленно сохраняя их структуру, не производя гомогенизацию.

В работе также был произведен обзор современного процесса обработки гетерогенных данных, показаны основные технологические подходы на каждом из этапов процесса.

Литература

1. Гомогенные и гетерогенные распределенные базы данных [Электронный ресурс]. – Режим доступа: https://studfiles.net/preview/5584820/page:31/ (дата обращения: 04.01.2019).
2. Технологии слияния данных (Data fusion) [Электронный ресурс]. – Режим доступа: http://sa.technolog.edu.ru/files/ananchenko/2.pdf (дата обращения: 04.01.2019).
3. The right tool for the job: Azure Data Factory v2 vs. Integration services [Электронный ресурс]. – Режим доступа: https://www.blue-granite.com/blog/the-right-tool-for-the-job-azure-data-factory-v2-vs.-integration-services (дата обращения: 04.01.2019).

Капутерка Эвелина Александровна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4172
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: kap-ehvelina@yandex.ru

Смолин Артем Александрович
Год рождения: 1977
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.философ.н.
e-mail: smolin@corp.ifmo.ru

УДК 004.5

ВЛИЯНИЕ АНИМИРОВАННЫХ ПЕРЕХОДОВ В ГРАФИЧЕСКИХ ИНТЕРФЕЙСАХ НА ЧЕЛОВЕКО-КОМПЬЮТЕРНОЕ ВЗАИМОДЕЙСТВИЕ

Капутерка Э.А.
Научный руководитель – к.философ.н. Смолин А.А.

Работа выполнена в рамках темы НИР «Исследование влияния анимированных переходов на эффективность человеко-компьютерного взаимодействия».

В работе проведен анализ влияния анимированных переходов на человеко-компьютерное взаимодействие, приведены результаты некоторых исследований в данной области, раскрывающие различные аспекты этого влияния. Также коротко описаны предпосылки к возникновению интереса проектировщиков интерфейсов к использованию анимации в своей работе, и поднят вопрос о необходимости проведения дальнейших исследований в данной области.
Ключевые слова: анимированные переходы, анимация, интерфейс, человеко-компьютерное взаимодействие, навигационные переходы.

В настоящее время технологии стремительно развиваются. Появляется больше возможностей, открываются новые перспективы для создания наиболее эффективных человеко-компьютерных интерфейсов. Вместе с тем идет активное повсеместное внедрение компьютерных технологий. На данный момент наиболее часто встречающимся типом интерфейса «человек-машина» являются так называемые графические пользовательские интерфейсы.

Таким образом, открываются пути для поиска новых способов повышения качества человеко-компьютерного взаимодействия. В данном контексте актуальной для исследований областью является изучение влияния анимации на эффективность человеко-компьютерного взаимодействия.

Частным случаем анимации в интерфейсах являются анимированные переходы. Анимированный переход в контексте графических пользовательских интерфейсов представляет собой отображение преобразования между различными состояниями пользовательского интерфейса с помощью анимации, т.е. без резких визуальных изменений.

Анимированные переходы, а также остальные типы анимации в интерфейсе (анимация прокрутки, анимация загрузки и пр.) имеют потенциал для использования в целях повышения качества разрабатываемого интерфейса.

Все более активное обращение проектировщиков интерфейсов к анимации является отражением общих тенденций развития пользовательских интерфейсов. Так, одной из первых дизайн-концепций графических интерфейсов в визуальном плане был скевоморфизм, суть которого заключалась в том, чтобы преподнести цифровой материал таким же, как и его физический аналог. Однако такой подход к проектированию шел вразрез с основным принципом дизайна полезного действия – как можно меньше интерфейса, т.е. меньше препятствий и лишних деталей на пути к достижению пользовательских целей. Поэтому постепенно появилась и была практически повсеместно внедрена другая концепция, заключающаяся в отказе от полного подражания элементов интерфейса объектам реального мира. Со временем были сформированы основные принципы «плоского дизайна». А в гайдлайне «Material design» уже можно отметить наличие таких элементов, как: тени, анимации. Повсеместное применение анимации может рассматриваться как новый виток в развитии графических пользовательских интерфейсов, который способен сгладить тот недостаток полезной информации, который образуется при использовании элементов «плоского дизайна».

Чтобы наиболее полно ответить на вопрос о влиянии анимированных переходов на человеко-компьютерное взаимодействие, нужно знать, как анимация способна менять качество интерфейса.

Так, анимация в интерфейсе может:

- обозревать, что было сделано;
- показать, что можно сделать;
- показать, чего нельзя сделать;
- показать, как можно делать;
- подсказать, как не нужно делать.

Если говорить об анимированных переходах, то основной их функцией является отображение связи между экранами, что делает пользовательский опыт более плавным.

Каждый раз, когда пользовательский опыт прерывается, вероятность ухода пользователя увеличивается, он не достигает своей цели. Переключение с одной страницы на другую часто приводит к этому прерыванию, показывая белый экран без содержимого, слишком долгую загрузку или иным образом выводя пользователя из контекста, в котором он находился до открытия новой страницы.

Переходы между страницами могут улучшить восприятие, сохраняя (или даже улучшая) контекст пользователя, поддерживая его внимание и обеспечивая визуальную непрерывность и положительную обратную связь. В то же время переходы по страницам также могут быть эстетически приятными [1].

Ответ на вопрос о том, почему анимированные переходы облегчают понимание событий, происходящих в интерфейсе, кроется в самом человеческом восприятии. Когда пользователь не может визуально отследить происходящие в интерфейсе изменения, ему сложнее понять причинно-следственную связь между ушедшим и появившимся экраном интерфейса. Чтобы сделать это, человеку необходимо сравнить текущее состояние и предыдущее, а затем выявить разницу между ними. Анимация же, в частности анимированный переход, может позволить пользователю думать непосредственно о выполнении своих задач, а не о том, почему сменился тот или иной экран интерфейса [2]. Это может способствовать и уменьшению количества ошибок при выполнении задач. Ничто не кажется более неестественным, чем внезапное изменение, потому что внезапные изменения не существуют в реальном мире.

Анимация способна улучшить юзабилити без использования тяжеловесных графических решений. Она создает ощущение работы с объектом реального мира, оставляя минимум

интерфейса, делая его легким. Таким образом, грамотное использование анимации позволяет создать иллюзию – взаимодействие с виртуальным интерфейсом ощущается человеком, как работа с привычным физическим объектом. При этом количество лишних элементов интерфейса уменьшается.

В первую очередь преимущества анимации связаны с такой психической функцией человеческого сознания, как кратковременная память. Анимация какого-либо изменения в интерфейсе – например, скрытия информационного блока на сайте – позволяет «продлить» в сознании пользователя факт изменения и зафиксировать его с большей надежностью, а также придать факту изменения дополнительные смыслы.

Компания Ник Бабич в статье [3] выделяет две роли анимированных переходов:

– определение пространственных отношений между экранами и элементами;
– избегание неожиданного перехода, помощь пользователям в понимании изменений, которые произошли в интерфейсе.

Помимо улучшения понимания структуры интерфейса, а также причинно-следственных связей между действием, вызвавшим изменение в интерфейсе, и самим произошедшим изменением, анимированные переходы оказывают влияние и на общее восприятие интерфейса пользователями.

Так, ряд исследований подтверждают то, что использование анимации и, в частности, анимированных переходов может способствовать повышению уровня субъективной удовлетворенности интерфейсом.

Например, в работе С. Дайера и Н. Адамо-Виллани [4] описывается эксперимент, по результатам которого анимированный интерфейс получил более высокие оценки по эстетической привлекательности, нежели статичный.

По результатам некоторых исследований можно сделать и вывод о том, что анимированные переходы влияют и на запоминание структуры интерфейса, т.е. расположения экранов относительно друг друга, их пространственных отношений.

Одним из таких исследований является работа Б. Бедерсона и А. Болтман [5]. Авторы пришли к выводу, что анимированные переходы улучшили способность пользователей воспринимать расположение экранов друг относительно друга, что говорит о лучшем понимании структуры интерфейса. Испытуемые лучше запомнили структуру того интерфейса, в котором переход между состояниями осуществлялся с помощью анимированных переходов.

Стоит отметить и тот факт, что различные анимированные переходы могут оказывать разное влияние на человеко-компьютерное взаимодействие. Например, длительность анимированных переходов оказывает влияние на восприятие интерфейса, может вызывать разные эмоции у пользователя. Слишком медленная анимация, например, при раскрытии плашек с информацией, может вызывать раздражение пользователя. Также важно и то, с помощью какого устройства происходит взаимодействие.

Таким образом, актуальность данной темы обусловлена развитием технологий и необходимостью поиска организации наиболее эффективного человеко-компьютерного взаимодействия. Анимация, в общем, и анимированные переходы, как частный ее случай, являются одним из недостаточно изученных аспектов графического интерфейса, который способен улучшить человеко-компьютерное взаимодействие, повысить уровень его эффективности. Именно поэтому необходимо проводить дальнейшие исследования в данной области.

Литература

1. Irani P., Shanmugasundaram M. The Effect of Animated Transitions in Zooming Interfaces // AVI. – 2008. – P. 396–399.
2. Chang B.-W., Ungar D. Animation: From Cartoons to the User Interface // UIST’93: User Interface Software and Technology. – 1993. – P. 45–55.

3. Babich N. XD Essentials: How Functional Animation Helps Improve UX [Электронный ресурс]. – Режим доступа: https://theblog.adobe.com/xd-essentials-how-functional-animation-helps-improve-ux/ (дата обращения: 04.01.2019).
4. Adamo-Villani N., Dyer S. Animated Versus Static User Interfaces: A Study of Mathsigner™ // International Journal of Computer and Information Engineering. – V. 2(2). – 2008. – P. 399–404.
5. Bederson B.B., Boltman A. Does Animation Help Users Build Mental Maps of Spatial Information? // HCIL Technical Report. – 1998. – P. 28–35.

Кинцель Никита Борисович
Год рождения: 1999
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р3273
Направление подготовки: 44.03.04 – Профессиональное обучение
e-mail: nikita.kintsel@yandex.ru

Шаров Даниил Дмитриевич
Год рождения: 1995
ООО «Оптимус», технический специалист
e-mail: Danyelsharov@gmail.com

Мамутова Людмила Артемьевна
Год рождения: 1951
Университет ИТМО, факультет программной инженерии
и компьютерной техники, тьютор
e-mail: mamutoval@mail.ru

УДК 7.017.9

ИЛЛЮЗИИ ВОСПРИЯТИЯ ПЕРСПЕКТИВЫ

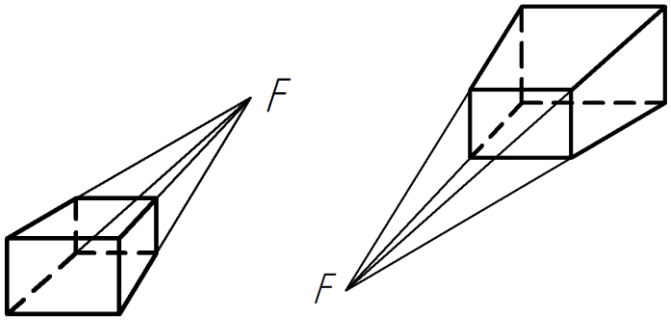
Кинцель Н.Б., Шаров Д.Д.
Научный руководитель – Мамутова Л.А.

В работе представлена модель, демонстрирующая иллюзию восприятия перспективы на примере архитектурного объекта.
Ключевые слова: иллюзия, перспектива, обратная перспектива.

В процессе индивидуального развития человека и приобретения им визуального опыта формируется восприятие размеров удаленных объектов как постоянных, несмотря на то, что человек видит их в перспективе. Чем объект более удален от наблюдателя, тем кажется он меньше. Однако в архитектуре издавна использовались приемы зрительного искажения пространства с целью создать видимость простора и глубины или наоборот видимости сжатости и замкнутости [1, 2]. Иногда создавались необычные эффекты просто для красоты или развлечения зрителей.

Задачей данной работы являлось создание физической модели, демонстрирующей один из таких необычных эффектов на примере архитектурного объекта. В качестве объекта создания модели было выбрано здание Баптистерия Сан-Джованни во Флоренции.

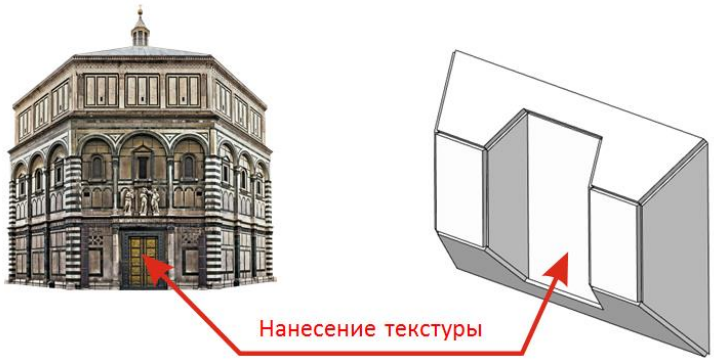
Согласно теории линейной перспективы, параллельные прямые линии пересекаются в бесконечно удаленной точке (точке схода), расположенной за объектом. На рис. 1, а, эта точка обозначена буквой *F*. Если искусственно переместить эту точку в пространство между зрителем и объектом, возникнет оптическая иллюзия, так называемая обратная перспектива (рис. 1, б).



а б

Рис. 1. Перспектива: линейная (а) и обратная (б)

Такое искусственное расположение точки схода предполагалось достичь за счет размещения текстур ближних граней на соответствующих гранях модели, выполненных специально заглубленными.



а б

Рис. 2. Схема нанесения текстуры: объект (а) и 3D-модель (б)

Для объекта (рис. 2, а) были созданы: 3D-модель (рис. 2, б) в графическом редакторе AutoCAD и развертка в графическом редакторе Adobe Photoshop CC (рис. 3). Грани модели были вырезаны из поливинилхлорида на гравировальном станке в ФабЛабе Технопарка Университета ИТМО и склеены. Распечатанные части развертки наклеивались на грани модели.

Рис. 3. Развертка текстуры

Далее проверялось наличие эффекта путем перемещения наблюдателя от центра объекта вправо и от центра влево (рис. 4). При перемещении наблюдателя влево раскрывалась правая сторона (рис. 4, б), а при перемещении вправо раскрывалась левая сторона (рис. 4, в). Для непосвященного наблюдателя, рассматривающего сначала центральный вид и ожидающего картину прямой перспективы, когда при смещении влево открывается левая сторона, а при смещении вправо – правая, полученный нами эффект должен стать неожиданным и занимательным.

а б в

Рис. 4. Изменение вида при перемещении зрителя: вид при смещении наблюдателя влево – раскрывается правая сторона (а); вид по центру объекта (б); вид при смещении наблюдателя вправо – раскрывается левая сторона (в)

Данная модель и возникающий эффект были продемонстрированы в Шанхайском музее науки и технологии на выставке Magic of light (Магия света) в 2019 году.

Заключение. Была создана модель, демонстрирующая иллюзию восприятия перспективы, подтверждающая, что при определенных условиях, а именно при размещении ближайших к зрителю плоскостей на отдаленные грани модели, возникает иллюзия восприятия перспективы из-за смещения точки схода в пространство между зрителем и объектом.

Литература

1. Калащникова О.Б., Горовенко Л.А. Использование оптических иллюзий в архитектуре и строительстве // Международный студенческий научный вестник. – 2016. – № 5(3). – С. 355–358.
2. Маркузон В.Ф. Всеобщая история архитектуры в XII томах. – Том V. Архитектура Западной архитектуры XV–XVI веков. – М.: Стройиздат, 1967. – 675 с.

Ключев Владислав Аркадьевич
Год рождения: 1997
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41002
Направление подготовки: 09.04.01 – Вычислительные системы и сети
e-mail: vakluchev@gmail.com

Жовницкий Владислав Альбертович
Год рождения: 1994
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника

Ястребов Егор Евгеньевич
Год рождения: 1994
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41101
Направление подготовки: 09.04.04 – Информационно-вычислительные системы
e-mail: astore.hawk@gmail.com

Ключев Аркадий Олегович
Год рождения: 1967
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: kluchev@gmail.com

УДК 004.431

НИЗКОУРОВНЕВОЕ ПРОГРАММИРОВАНИЕ АППАРАТНОГО ОБЕСПЕЧЕНИЯ И ЕГО МЕСТО В СОВРЕМЕННОМ УЧЕБНОМ ПРОЦЕССЕ

Ключев В.А., Жовницкий В.А., Ястребов Е.Е.
Научный руководитель – к.т.н., доцент Ключев А.О.

В работе приведены проблемы низкоуровневого программирования аппаратного обеспечения и его места в современном учебном процессе, произведен анализ рынка труда, и предложены способы увеличения популярности данного направления.
Ключевые слова: киберфизическая система, bare-metal, embedded, вычислительная техника, язык программирования.

Низкоуровневое программирование аппаратного обеспечения (Bare-metal-programming) – это написание программного обеспечения, которое напрямую работает на аппаратном обеспечении без какой-либо базовой абстракции, такой как операционные системы. На момент выполнения данной работы популярность низкоуровневого программирования не очень

высока. Исходя из анализа требуемых компетенций в области IT на портале headhunter.ru можно подчеркнуть преобладание вакансий web-разработчиков и разработчиков мобильных приложений. Большинство таких вакансий не требуют наличие огромного опыта во владении тем или иным инструментом разработки. С вакансиями в области низкоуровневого программирования ситуация противоположная. Вакансий мало, требования к опыту очень высокие. Если же сравнивать заработную плату, то верхняя и нижняя планки примерно на одном уровне. Тогда сразу появляется вопрос – «Зачем мне изучать низкоуровневое программирование, если web-разработчик получает такую же зарплату, тратя гораздо меньше времени на освоение инструментов разработки?». Однако изучение «голого железа» дает важные знания в области вычислительной технике, понимание основных принципов работы позволит разработчику избегать неочевидных с точки зрения высокоуровневых языков программирования проблем, научит эффективно использовать ресурсы вычислительной машины (процессор и память). Позволит изучить модели вычислений, такие как: машина фон-Неймана, конечные автоматы, сети процессов Кана. Если углубиться еще и в схемотехнику, то можно освоить модель дискретных событий. Изучение низкоуровневого программирования (рис. 1, а) даст понимание таких понятий как разделение времени, параллелизм, критические секции, семафоры и гонки на простых примерах [1–7].

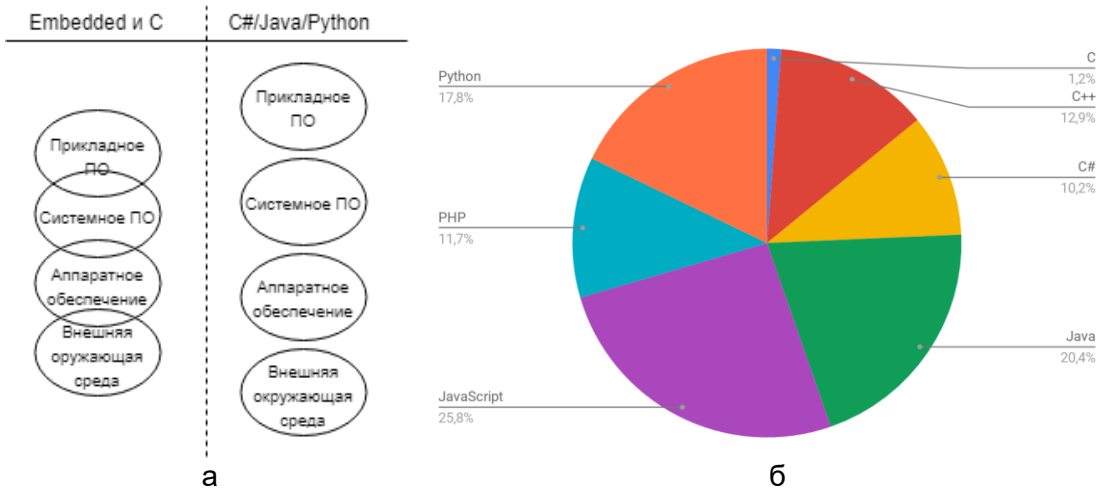
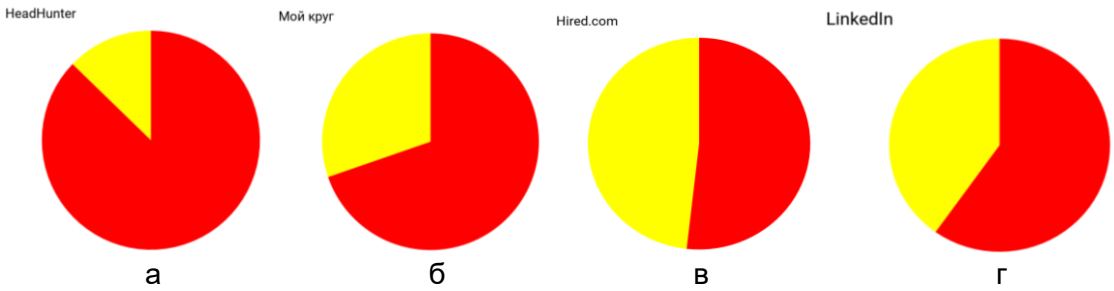


Рис. 1. Взаимное влияние уровней во встраиваемых системах и системах общего назначения (а); соотношение требуемых языков программирования на headhunter (б)

Кому будет полезно изучение bare-metal? В первую очередь абсолютно всем программистам, даже тем, кто использует такие языки программирования как Haskell или Scala: системным администраторам, разработчикам драйверов и модулей ядра для различных операционных систем, специалистам по разработке программного обеспечения (ПО) для бэкэнда, разработчикам коммуникационных систем, архитекторам программных систем, системным аналитикам и системным инженерам.

Результаты исследования состояния рынка труда получились следующими (рис. 1, б): на диаграмме можно увидеть соотношение наиболее востребованных языков программирования. Языки C и C++ в рамках низкоуровневого программирования занимают менее 15% от общей массы, что говорит в невысокой популярности и небольшом спросе на рынке.

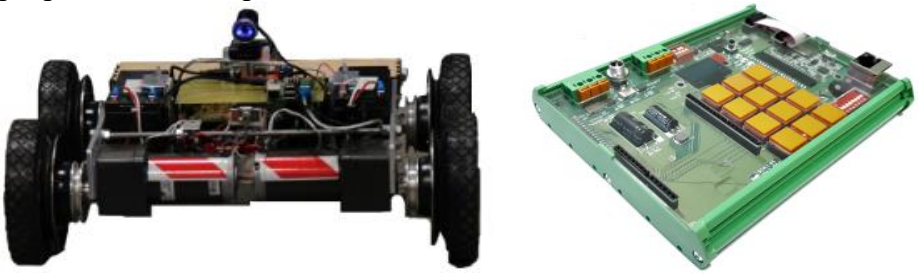
Также был проведен анализ и сравнение нашего и зарубежного рынков вакансий в рамках популярных российских порталов HeadHunter, Мой круг и зарубежных Hired.com, LinkedIn на примере двух вакансий: Embedded software developer и Java software developer. На рис. 2 представлены 4 диаграммы. Если на рис. 2, а, б, (Россия) заметно сильное преобладание Java software developer, то на рис. 2, в, г, (США и Канада) эти вакансии находятся в балансе.



а б в г

Рис. 2. Соотношение вакансий Embedded software developer (желтый) и Java software developer (красный): HeadHunter (а); Мой круг (б); Hired.com (в); LinkedIn (г)

В рамках исследования киберфизических систем был разработан мобильный фреймворк на базе колесного робота, а также совместно с дизайн центром ЛМТ разработан новый модульный учебный лабораторный комплекс SDK-1.1M (рис. 3). На их примере можно раскрыть и показать студентам все аспекты киберфизической системы и вычислительной техники в целом, не выходя за рамки одного устройства. Это позволит сделать низкоуровневое программирование более интересным не только для студентов этого направление, но и студентам-программистам в принципе.



а б

Рис. 3. Мобильная платформа (а), учебный стенд SDK-1.1M

В качестве решения проблемы проблемы предлагается сделать следующее: дать студентам, изучающим web-технологии и программирование высокого уровня специальный курс, позволяющий познакомиться с программированием встроенных систем. Этот курс позволит студентам:

1. научаться писать код эффективно и надежно;
2. научиться учитывать ограниченность ресурсов устройства;
3. научиться решать такие проблемы как утечка памяти;
4. разобраться в параллельных процессах, разделении времени, семафорах и гонках.

В дальнейшем планируется создать подробные методические указания с примерами на C и Java для того, чтобы показать основные «болевые точки» высокоуровнего программирование на простых примерах.

Литература

1. Lee E.A., Seshia S.A. Introduction to Embedded Systems – A Cyber-Physical Systems Approach Second Edition. – MIT Press, 2017. – 564 p.
2. Московская биржа. Индекс МосБиржи и индекс РТС [Электронный ресурс]. – Режим доступа: https://www.moex.com/ru/index/RTSI (дата обращения: 04.01.2019).
3. Хэдхантер [Электронный ресурс]. – Режим доступа: http://hh.ru (дата обращения: 04.01.2019).
4. Мой круг [Электронный ресурс]. – Режим доступа: https://moikrug.ru (дата обращения: 04.01.2019).
5. Hired [Электронный ресурс]. – Режим доступа: http://hired.com (дата обращения: 04.01.2019).
6. LinkedIn [Электронный ресурс]. – Режим доступа: http://linkedin.com (дата обращения: 04.01.2019).
7. Хэррис Д., Хэррис С. Цифровая схемотехника и архитектура компьютера. – Morgan Kaufman, 2013. – 1627 с.

Жовницкий Владислав Альбертович
Год рождения: 1994
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника

Ключев Владислав Аркадьевич
Год рождения: 1997
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41002
Направление подготовки: 09.04.01 – Вычислительные системы и сети
e-mail: vakluchev@gmail.com

Ястребов Егор Евгеньевич
Год рождения: 1994
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41101
Направление подготовки: 09.04.04 – Информационно-вычислительные системы
e-mail: astore.hawk@gmail.com

Ключев Аркадий Олегович
Год рождения: 1967
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: kluchev@gmail.com

УДК 004.4

ПРОБЛЕМЫ УДАЛЕННОГО ОБНОВЛЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КИБЕРФИЗИЧЕСКИХ СИСТЕМ

Жовницкий В.А., Ключев В.А., Ястребов Е.Е.
Научный руководитель – к.т.н., доцент Ключев А.О.

В работе приведены проблемы удаленного обновления и конфигурирования программного обеспечения киберфизических систем, произведена их оценка и приведены способы решения.
Ключевые слова: киберфизическая система, программная система, информационная система, обновление ПО, холонические системы, холархии.

Киберфизические системы (Cyber-Physical System, CPS) – это системы, представляющие собой общности связанных между собой коммуникациями реальных, физических систем, виртуальных систем и их контроллеров [1]. Основным принципом при разработке таких систем представляется идея о том, что CPS жестко зависят от связи вычислительных процессов и процессов реального мира, т.е. природных, физических. Внесение в систему

физических объектов подвергает ее воздействию со стороны огромного количества важнейших параметров природного объекта (примером таких объектов является контроллер уличного освещения, в разработку которого входит учет зависимости исправной работы этого контроллера от температуры, влажности и многих других параметров). Вычисления, производимые компьютерной системой, которая управляет физическим устройством, зависят от состояний этого устройства и наоборот. CPS можно рассматривать в рамках холонического подхода в терминах холонов – систем, являющихся целостными, которые, в свою очередь, являются частью большего – холархий, иерархий холонов [2].

При усложнении системы увеличиваются расходы на ее содержание, а также необходимо больше людей с соответствующей квалификацией для сопровождения такой системы. Одной из важнейших проблем CPS является удаленное обновление программного обеспечения (ПО) таких нетривиальных систем.

Современные CPS очень сложны, разнообразны и имеют множество различных зависимых подсистем [3]. Такое масштабное ПО невозможно реализовать за одну итерацию разработки, запустить ее в реальном мире и навсегда забыть про нее. На отдельные части CPS могут оказывать влияние миллионы природных и человеческих факторов, которые невозможно все воспроизвести в тестовых условиях при создании системы или при ее проверке и приемке.

Следовательно, любая CPS должна иметь проверенный и надежный механизм обновления. Однако обновление сложных распределенных систем, а особенно киберфизических – нетривиальная задача. Обновление CPS требует длительной подготовки, которая может занимать месяцы, и принятия во внимание многих параметров [4].

Одной из трудностей обновления CPS является ее труднодоступность [5]. Система целиком или ее части могут быть расположены далеко от освоенных человеком мест, там, где практически отсутствует какая-либо связь с остальным миром. На рис. 1 приведена карта покрытия мобильными операторами между Новосибирском и Кемерово. На карте видно, что даже здесь не везде есть мобильная связь.

Рис. 1. Карта покрытия мобильными операторами

Следующей проблемой следует отметить человеческий фактор и отсутствие специалистов должного уровня для произведения ручного обслуживания и обновления системы.

Обновление системы может прийти искаженным, сломаться при передаче из-за плохого качества связи, т.е. при любом неудачном стечении обстоятельств при обновлении системы возникает риск полного отказа системы.

Авторы в работе стремились к минимизированию человеческого воздействия в работе системы, т.е. ее работа, обслуживание и обновление должно происходить в автономном режиме.

Одним из вариантов решения проблем удаленного обновления киберфизических систем является адаптивный алгоритм обновления. Суть работы данного алгоритма заключается в нескольких принципах:

- сбор статистики обновлений;
- отслеживание параметров обновлений и системы в зависимости от версии ПО;
- система автоматической оценки эффективности работы прошивок;
- система отката к наиболее эффективной прошивке;
- модульность прошивки (выделение критических для работы системы элементов).

Для этого было предложено разработать систему автоматического получения, обновления и переключения прошивок на лету. Среди основных задач поставлены:

- разработка модели оценки успешности прошивки по различным критериям (например, одна прошивка имеет больше успехов по выполнению целевой функции, другая имеет отказоустойчивость при получении и передаче данных по сети);
- разработка механизма работы с прошивками: механизм должен включать в себя возможность переключить прошивку на лету при необходимости или при возникновении ошибок в текущей.

Таблица. Коэффициенты качества версий прошивки ПО

№	Версия прошивки ПО	Качество связи, %	Обновление, %	ЦФ1, %	ЦФ2, %	ЦФ3, %	КВ
1	1.0.0	80	94	0	22	62	173,4
2	1.0.1	30	85	20	35	30	146,75
3	1.0.2	78	94	85	92	96	293,4
.		.	.	.	.	.	.
.		.	.	.	.	.	.
N	n.0.0	X	.	.	.	.	.

В таблице представлены коэффициенты качества версии прошивки ПО, рассчитанные исходя из скоринга по критически важному для системы функционалу, который, в свою очередь, рассчитывается исходя из соотношения количества успешных выполнений программы к количеству сбоев и ошибок, вызванных во время работы, где ЦФ(N) – это целевая функция, например, расчет времени переключения реле контроллером, а КВ – коэффициент качества версии, который рассчитывается с учетом вклада каждого пункта оценки версии (качество связи, обновление и т.д.).

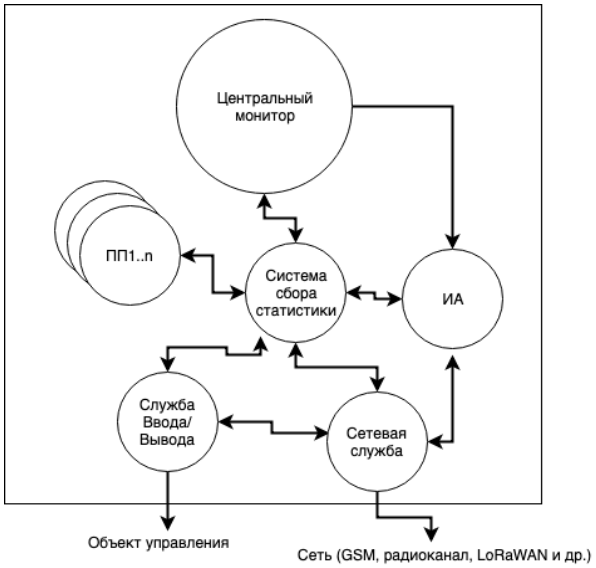


Рис. 2. Удаленное обновление CPS (сеть процессов Кана)

На рис. 2 показан механизм работы алгоритма. ПП1..*n* – набор прикладных программ, ИА – инструментальный агент, способный перепрограммировать элементы системы, центральный монитор – служба, принимающая решения на основании данных, полученных от системы сбора статистики, и передающая указания по выбору прошивки инструментальному агенту.

Поскольку подобные алгоритмы обновления систем сложны для тестирования и отладки на реальных системах, велика вероятность в результате ошибочных обновлений получить заблокированную и не отвечающую систему. Для возможности проведения экспериментов и тестирования системы обновления и алгоритмов был разработан мобильный фреймворк (рис. 3) на основе колесного робота с несколькими критическими для работы обновляемыми модулями.

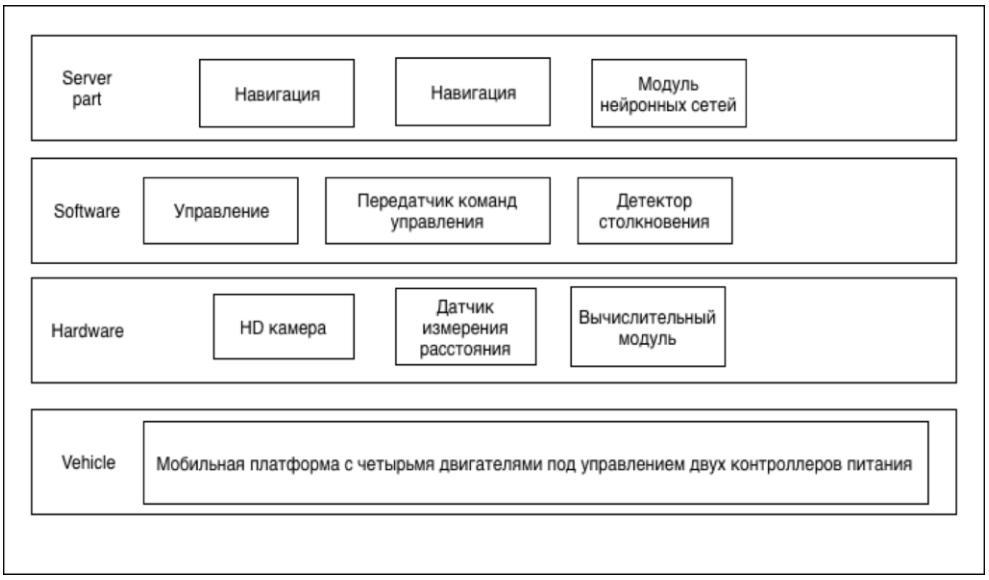


Рис. 3. Архитектура платформы

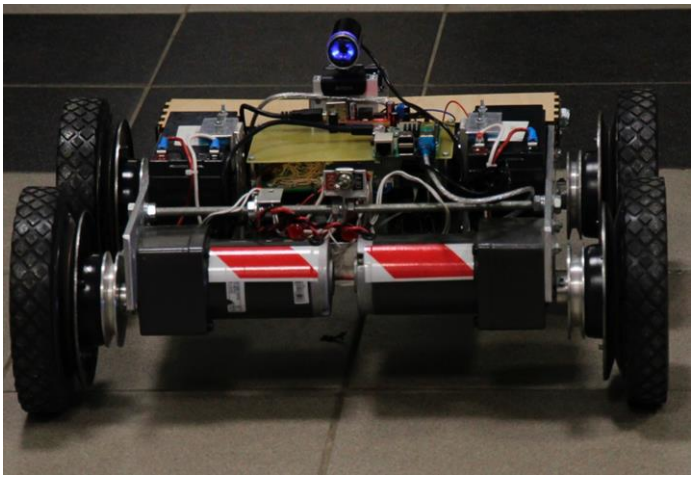


Рис. 4. Мобильная платформа на базе колесного робота

Мобильная платформа (рис. 4) позволяет имитировать мобильную CPS (например, курьер, беспилотный автомобиль), а также имеет достаточные вычислительные мощности для запуска нейронных сетей и генетических алгоритмов, позволяющих реализовывать предлагаемую систему удаленного обновления CPS.

В результате работы были исследованы проблемы удаленного обновления CPS, предложен вариант их решений, разработан алгоритм удаленного обновления CPS и сконструирована мобильная платформа для тестирования и отладки данного алгоритма.

Литература

1. Lee E.A. Cyber physical systems: Design challenges // 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC). – 2008. – P. 363–369.
2. Wang L., Haghighi A. Combined strength of holons, agents and function blocks in cyber-physical systems // Journal of manufacturing systems. – 2016. – V. 40. – P. 25–34.
3. Ястреб Н.А. Индустрия 4.0: киберфизические системы, разумное окружение, Интернет вещей [Электронный ресурс]. – Режим доступа: https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwi7obXU6_DkAhUGxYsKHe-6A50QFjABegQIBRAB&url=https%3A%2F%2Fdocplayer.ru%2F28319747-Industriya-4-0-kiberfizicheskie-sistemy-razumnoe-okruzhenie-internet-veshchey.html&usg=AOvVaw24c2MfN7Sow_1W3hSoGTm2 (дата обращения: 16.12.2018).
4. Полтавец А.И., Петров И.П. Безопасность киберфизических систем // Сборник материалов XV Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых. – С. 46.
5. Белокуров С.В., Багринцева О.В., Змеев А.А. Особенности принятия решений при обеспечении информационной безопасности удаленных объектов // Проблемы обеспечения безопасности при ликвидации последствий чрезвычайных ситуаций. – 2012. – № 1. – С. 269–271.

Ковалев Василий Васильевич
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P4263
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: vasya.kowalev@yandex.ru

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.57

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ ОБСЛУЖИВАНИЯ И РАЗВЕРТЫВАНИЯ ОБЛАЧНЫХ РЕШЕНИЙ В ВЕБЕ

Ковалев В.В.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

В работе приведен сравнительный анализ моделей обслуживания и развертывания облачных решений. Описаны критерии проведений сравнительного анализа для каждой группы моделей. Для моделей обслуживания выявлены наиболее популярные и наиболее быстроразвивающиеся модели, а также описаны причины этого.
Ключевые слова: облачные технологии, облака, веб, модели обслуживания, модели развертывания.

В настоящее время облачные технологии прочно вошли в нашу жизнь. Множественный доступ к редактированию онлайн-документов, создание веб-приложений, облачные хранилища данных – вот далеко не полный список примеров, которые еще десятилетие назад были просто задумкой, а теперь стали привычным явлением.

Существует множество моделей предоставления облачных вычислений в вебе, и для разработки и публикации в Интернете своего проекта необходимо понимать различия между ними. Это позволит на ранних этапах определиться с требующимися ресурсами, инструментами и сервисами с учетом ограничений каждой из моделей.

Анализ интернет-источников показал, что, хотя и есть большое количество сравнительных анализов различных моделей (например, моделей обслуживания), зачастую сравнение производится на одной и той же выборке (например, для моделей обслуживания это SaaS, PaaS и IaaS). Обычно для остальных моделей приводится только описание, например, на сайтах поставщиков или в статьях передовых исследователей [1]. Таким образом, проводимое исследование позволит восполнить данный пробел.

Целью работы являлось проведение сравнительного анализа моделей обслуживания и развертывания как с экономической, так и с технической точек зрения. В задачи входит определение критериев сравнения моделей и непосредственно проведение сравнительного анализа.

В исследовании [2] был проведен обзор моделей предоставления облачных вычислений в вебе, даны определения основным понятиям в этой области, а также описаны основные отличия моделей друг от друга, механизмы их работы и приведены примеры использования

данных моделей в вебе. Данное исследование является базой для проведения настоящего сравнительного анализа. Вначале авторами был проведен сравнительный анализ моделей обслуживания, а затем – моделей развертывания.

Модели обслуживания различаются по предоставляемому потребителю ресурсу или комплекту ресурсов. Для того чтобы понять, в каком случае необходимо использовать ту или иную модель обслуживания, а также какие модели наиболее перспективны с точки зрения капиталовложения, необходимо оценить достоинства и недостатки каждой из них, т.е. произвести их сравнение. Чтобы это сделать, необходимо определиться с критериями сравнения.

Для бизнеса важными являются такие критерии, которые позволяют, во-первых, сделать вывод о полезности (в виде увеличения дохода, упрощения управления ресурсами и т.д.) той или иной технологии, во-вторых, оценить перспективы ее развития, чтобы затраты успели окупиться до выхода технологии из употребления. На основании этого были определены следующие критерии сравнения моделей обслуживания:

- что предоставляет (какие ресурсы предоставляются в пользование при заказе той или иной услуги);
- занимаемая доля рынка (доля рынка и ее стоимость по последним имеющимся данным);
- динамика развития (планируемая занимаемая доля рынка и среднегодовой прирост стоимости).

В таблице представлено сравнение нескольких моделей обслуживания по определенным выше критериям. Для оценки динамики развития используется CAGR («Compound Annual Growth Rate», совокупный среднегодовой темп роста). Он вычисляется по формуле:

$$CAGR = \frac{V(t_n)^{\frac{1}{t_n - t_0}}}{V(t_0)} - 1. \quad (1)$$

где $t_n - t_0$ – разность лет оценки темпа роста; $V(t_0)$ – стоимость в момент времени t_0, млрд долл.; $V(t_n)$ – предполагаемая стоимость в момент времени t_n, млрд долл.

Таблица. Сравнение моделей обслуживания

Модель обслуживания (X as a Service)	Что предоставляет в пользование	Занимаемая доля рынка, млрд долл.	Динамика рынка, млрд долл.
Software (SaaS)	Приложение	58,6 (2017 г.) [3]	99,7 (2020 г.) CAGR: +19,38%
Platform (PaaS)	Среда выполнения, базы данных (БД), гостевые операционные системы (ОС), контейнеры	11,4 (2017 г.) [3]	20,8 (2020 г.) CAGR: +22,19%
Infrastructure (IaaS)	Хранилище данных, основная ОС	34,7 (2017 г.) [3]	72,4 (2020 г.) CAGR: +27,78%
Backend (BaaS)	Среда выполнения, управление push/e-mail/SMS-уведомлениями, моделирование и управление данными, управление пользователями и правами доступа для них, статистика/аналитика	0,79 (2014 г.) [4]	28,1 (2020 г.) CAGR: +81,35%
Backup (BaaS)	Хранилище с копией состояния данных	1,3 (2017 г.) [5]	4,13 (2022 г.) CAGR: +26,01%
Disaster Recovery (DRaaS)	Хранилище с копией состояния инфраструктуры	1,72 (2016 г.) [6]	12,54 (2022 г.) CAGR: +39,25%

Модель обслуживания (X as a Service)	Что предоставляет в пользование	Занимаемая доля рынка, млрд долл.	Динамика рынка, млрд долл.
Database (DBaaS)	Базы данных, API для работы с БД	1,07 (2014 г.) [7]	14,05 (2019 г.) CAGR: +67,36%
Desktop (DaaS)	Стандартизированное рабочее место	7,08 (2016 г.) [8]	13,45 (2022 г.) CAGR: +11,29%
Network (NaaS)	Управление связью разных компонентов инфраструктуры между собой, управление пропускной способностью канала передачи данных	38,6 (2016 г.) [9]	59,38 (2021 г.) CAGR: +9,00%

На основании данных из таблицы можно сделать следующие выводы:

- наибольшую долю рынка занимают такие модели обслуживания, как SaaS, NaaS и IaaS;
- ввиду простоты и скорости создания облачных решений, а также развитого инструментария для этого, SaaS занимает наибольшую долю рынка на данный момент и продолжает развиваться;
- NaaS популярен в виду того, что позволяет легко и быстро связывать устройства по всему миру в единую систему, создавая распределенные структуры данных;
- IaaS также занимает большую долю рынка, так как предоставляет больше возможностей (большее количество уровней абстракции) для создания приложений. Кроме того, можно комбинировать более мелкие блоки, такие как BaaS, DBaaS, DRaaS и тому подобное, для создания наиболее подходящего и экономически выгодного облачного решения, а также предоставлять другим пользователям платформу для их разработок. Например, имея IaaS, можно предоставить ресурсы для создания PaaS- или SaaS-моделей;
- наиболее быстроразвивающимися моделями обслуживания являются BaaS, DBaaS и DRaaS;
- BaaS (Backend as a Service) становится все более популярным за счет того, что упрощает разработку наиболее сложной части приложения. Такое упрощение позволяет создавать облачные решения более быстро, что ведет к более быстрому получению дохода;
- также одной из самых сложных частей приложения является БД. Сложность заключается в том, что БД, спроектированная не оптимальным образом, усложняет работу с ней и приводит к задержкам в работе приложения, что, в свою очередь, негативно влияет на его популярность и посещаемость. DBaaS решает эту проблему и позволяет упростить разработку, развертывание, хранение и доступ к БД;
- DRaaS становится популярен потому, что более комплексно подходит к восстановлению: восстанавливаются не только данные, но и приложение. Особенно рост популярности заметен на контрасте с развитием Backup as a Service. Можно предположить, что за счет преимуществ DRaaS, его популярность со временем будет только расти, а BaaS – снижаться.

Далее перейдем к сравнительному анализу моделей развертывания. В источнике [10] приведены критерии и результаты комплексного сравнения трех моделей развертывания облаков: частного, публичного и гибридного. Общественное облако не рассматривается здесь по причине того, что оно создается для совместной работы нескольких организаций (что является необязательным), а в статье упор делается на выбор модели развертывания для использования именно одной организацией для своих нужд. Конечно, если есть необходимость в увеличении ресурсов, то они могут заимствоваться из общественного облака, но это уже происходит опосредованно и является областью ответственности поставщика, если облако не частное.

В работе также указываются проблемы, которые поспособствовали ее написанию. С одной стороны, при построении инфраструктуры «на местах» нет четкого понимания, какие нагрузки необходимо выносить в публичное облако. С другой стороны, все еще актуальна проблема выбора между частным и публичным облаком.

Далее приведены сравнения трех типов облаков. Всего было указано 23 критерия для оценки, разделенных на следующие группы:

- требования к приложению;
- обеспечение безопасности;
- операционные задачи;
- затраты.

Оценки представляют собой числа, кратные 0,25, которые определяют, насколько поддерживается тот или иной критерий для каждой модели развертывания. Чтобы понять в целом, какая модель развертывания наиболее оптимальна, подсчитаем количество баллов N для каждой модели развертывания (формулы (2)–(4)):

- для частного облака:

$$N = 11 \cdot 1 + 11 \cdot 0{,}75 + 0 \cdot 0{,}5 + 0 \cdot 0{,}25 + 1 \cdot 0 = 19{,}25; \quad (2)$$

- для публичного облака:

$$N = 6 \cdot 1 + 0 \cdot 0{,}75 + 6 \cdot 0{,}5 + 5 \cdot 0{,}25 + 6 \cdot 0 = 10{,}25; \quad (3)$$

- для гибридного облака:

$$N = 13 \cdot 1 + 9 \cdot 0{,}75 + 0 \cdot 0{,}5 + 0 \cdot 0{,}25 + 1 \cdot 0 = 19{,}75. \quad (4)$$

Согласно рассчитанным средним оценкам, наиболее оптимальным вариантом является модель гибридного облака. Это подтверждает статистика использования этих типов облаков [11]. Однако и модель частного облака также является очень распространенной, потому как порой конфиденциальность и целостность данных, а также независимость от сторонних поставщиков важнее, чем затраты на закупку дополнительных мощностей.

В результате работы был проведен сравнительный анализ моделей обслуживания и развертывания. Сравнительный анализ был проведен как с экономической точки зрения (занимаемая доля рынка, среднегодовой прирост), так и с технической (какие ресурсы для управления доступны заказчику). Данное исследование может послужить основой для проведения дальнейших исследований в этой области, например, проведения сравнительного анализа имеющихся на данный момент на рынке облачных сервисов, использующих затронутые выше модели обслуживания/развертывания.

Литература

1. Sugam Sharma. Evolution of as-a-Service Era in Cloud [Электронный ресурс]. – Режим доступа: https://arxiv.org/ftp/arxiv/papers/1507/1507.00939.pdf (дата обращения: 04.01.2019).
2. Ковалев В.В. Обзор моделей предоставления облачных вычислений в вебе // Альманах научных работ молодых ученых Университета ИТМО. – 2018. – Т. 7. – С. 167–170.
3. Gartner Forecasts Worldwide Public Cloud Services Revenue to Reach $260 Billion in 2017 [Электронный ресурс]. – Режим доступа: https://www.gartner.com/newsroom/id/3815165 (дата обращения: 04.01.2019).
4. Mobile Backend as a Service Market by Operating System & Vertical Types – 2020 [Электронный ресурс]. – Режим доступа: https://www.marketsandmarkets.com/Market-Reports/mobile-backend-as-a-servicembaas-market-813.html (дата обращения: 04.01.2019).
5. Cloud Backup Market by Solution, Services & Provider – 2022 [Электронный ресурс]. – Режим доступа: https://www.marketsandmarkets.com/Market-Reports/cloud-backupmarket-92942042.html (дата обращения: 04.01.2019).

6. Disaster Recovery as a Service Market by Provider & Type – 2022 [Электронный ресурс]. – Режим доступа: https://www.marketsandmarkets.com/MarketReports/recovery-as-a-service-market-962.html (дата обращения: 04.01.2019).
7. Cloud Database and Database as a Service (DBaaS) Market worth $14.05 Billion by 2019 Survey [Электронный ресурс]. – Режим доступа: https://www.marketsandmarkets.com/PressReleases/cloud-database-as-a-servicedbaas.asp (дата обращения: 04.01.2019).
8. Desktop Virtualization Market by Type & Vertical – Survey [Электронный ресурс]. – Режим доступа: https://www.marketsandmarkets.com/Market-Reports/desktopvirtualization-market-137215705.html (дата обращения: 04.01.2019).
9. Managed Network Services Market by Types & Vertical – 2021 [Электронный ресурс]. Режим доступа: https://www.marketsandmarkets.com/MarketReports/managed-network-services-market-901.html (дата обращения: 04.01.2019).
10. Особенности выбора между частным, публичным и гибридным облаком [Электронный ресурс]. – Режим доступа: http://iaas-blog.itgrad.ru/tendencii/osobennosti-vybora-mezhdu-chastnym-publichnym-i-gibridnymoblakom/ (дата обращения: 04.01.2019).
11. Cloud Computing Trends: 2017 State of the Cloud Survey [Электронный ресурс]. – Режим доступа: https://www.rightscale.com/blog/cloud-industry-insights/cloudcomputing-trends-2017-state-cloud-survey (дата обращения: 04.01.2019).

Кондулукова Дарья Александровна
Год рождения: 1993
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4270
Направление подготовки: 09.04.02 – Мультимедиа-технологии
в искусстве театра, кино и телевидения
e-mail: dariakondulukova@mail.ru

Сизиков Валерий Сергеевич
Год рождения: 1939
Университет ИТМО, факультет программной инженерии
и компьютерной техники, д.т.н., профессор
e-mail: sizikov2000@mail.ru

УДК 004.932, 519.642.3

ПОВЫШЕНИЕ ТОЧНОСТИ ВОССТАНОВЛЕНИЯ ИСКАЖЕННОГО ИЗОБРАЖЕНИЯ ПУТЕМ ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ ИСКАЖЕНИЙ ПО ФУРЬЕ-СПЕКТРУ

Кондулукова Д.А.
Научный руководитель – д.т.н., профессор Сизиков В.С.

Работа выполнена в рамках темы НИР № 619296 «Разработка методов создания и внедрения киберфизических систем».

Развивается спектральный метод оценки параметров функции рассеяния точки в задаче восстановления искаженных (смазанных, дефокусированных) изображений. Метод основан на анализе Фурье-спектра искаженного изображения. Данный метод дает возможность оценить параметры функции рассеяния точки: угол θ и величину Δ смазывания изображения, а также размер ρ пятна дефокусирования изображения. Получены новые оценки параметров θ и Δ с использованием частоты Найквиста и оценка параметра дефокусирования ρ с использованием функции Бесселя. Приведены результаты применения данного метода для обработки изображений. Данный метод может быть использован для повышения точности восстановления смазанных и дефокусированных изображений путем их математической обработки с помощью устойчивых методов решения интегральных уравнений Фредгольма I рода (некорректная задача).
Ключевые слова: искажения изображений (смазывание, дефокусирование), функция рассеяния точки, параметры искажения, Фурье-спектр, интегральные уравнения, MATLAB.

Введение. Различные устройства регистрации изображений (УРИ), а именно, цифровые фотоаппараты, видеокамеры, следящие системы, телескопы, микроскопы и др. фиксируют изображения объектов – людей, животных, технических деталей, номерных знаков автомобилей, астрономических объектов, биологических микроорганизмов и т.д. В этом случае изображение может быть смазано (из-за сдвига УРИ или движения объекта) или дефокусировано (из-за ошибочной установки фокуса), а также зашумлено внешним или инструментальным шумом [1–3]. Данные искажения изображений могут быть устранены техническим путем – съемкой при неподвижном УРИ или/и объекта (в этом случае не будет смаза изображения), правильной установкой фокуса (не будет дефокусирования), а также съемкой в отсутствие шумов.

Однако не всегда благоприятные условия сопутствуют съемке. Примеры: съемка быстро летящего самолета (снимок может получиться смазанным) [3, С. 172], нечеткие первоначальные снимки астрономических объектов телескопом «Хаббл» [3, С. 105; 4] и т.д.

Изображения могут быть восстановлены математически – путем их математической и компьютерной обработки ([2, 3, 5, 6] и др.). Такой путь является существенным дополнением к задаче восстановления изображения, когда техническое восстановление проблематично.

Математическая формулировка задачи восстановления изображений. Задача устранения как смаза, так и дефокусирования изображения часто выполняется путем решения двумерного интегрального уравнения (ИУ) Фредгольма I рода типа свертки [2, 3, 5–9]:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x-\xi, y-\eta)\, w(\xi,\eta)\, d\xi\, d\eta = g(x,y) + \delta g\,, \qquad (1)$$

где h – функция рассеяния точки (ФРТ) или ядро ИУ, которая является, как правило, пространственно-инвариантной (разностной) функцией; w и g – распределение интенсивности по истинному и искаженному изображениям соответственно; δg – шум.

Целью данной работы являлось повышение точности решения ИУ (1) путем уточнения ФРТ, или ядра́ $h(x, y)$ ИУ (1).

Методы решения интегрального уравнения (1). Задача решения ИУ (1) является некорректной (существенно неустойчивой) [2, 3, 10]. Поэтому для его решения были использованы устойчивые методы – регуляризации Тихонова [2, 3, 5, 6], а также параметрической фильтрации Винера [2, 3, 5, 6].

Решение ИУ (1) методом регуляризации Тихонова с преобразованием Фурье (ПФ) равно $w_\alpha(x, y) = F^{-1}\left(W_\alpha(\omega_1, \omega_2)\right)$, где F^{-1} – обратное ПФ, $\alpha>0$ – параметр регуляризации, и $W_\alpha(\omega_1, \omega_2)$ – регуляризованный спектр (двумерное ПФ) решения, равный [3, 6]

$$W_\alpha(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)\, G(\omega_1, \omega_2)}{\left|H(\omega_1, \omega_2)\right|^2 + \alpha\,(\omega_1^2 + \omega_2^2)^p}\,, \qquad (2)$$

где $p \ge 0$ – порядок регулязации (обычно p=1, 2 или 3), $H(\omega_1, \omega_2) = F(h(x, y))$ и $G(\omega_1, \omega_2) = F(g(x, y))$ спектры (ПФ), причем F – прямое ПФ.

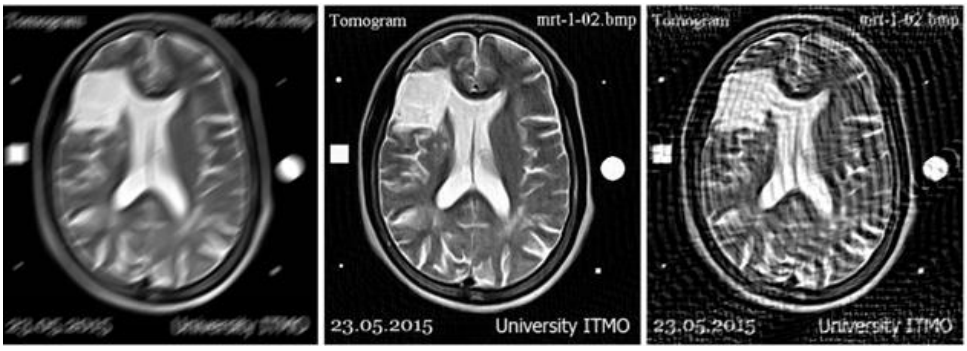
Решение ИУ (1) методом параметрической фильтрации Винера равно $w_K(x, y) = F^{-1}\left(W_K(\omega_1, \omega_2)\right)$, где $K \ge 0$ – параметр метода, равный отношению шум/сигнал по мощности (NSPR), и спектр решения равен [2, 3, 6]

$$W_K(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)\, G(\omega_1, \omega_2)}{\left|H(\omega_1, \omega_2)\right|^2 + K}. \qquad (3)$$

Решение ИУ (1) методами регуляризации Тихонова (РТ) и параметрической фильтрации Винера (ПФВ) реализовано в пакете IPT-системы MATLAB в виде m-функций deconvreg.m и deconvwnr.m соответственно [7]. Однако даже методы РТ и ПФВ весьма чувствительны к погрешностям параметров искажения – значениям Δ и θ смаза, а также значению ρ дефокусирования изображения, т.е. к неточностям знания ФРТ, или к погрешностям ядра́ h ИУ (1).

В качестве иллюстрации на рис. 1, а, приведено разработанное авторами [3, С. 14] изображение (томограмма-фантом) mrt-1-02.bmp 407×380 пкс, смазанное под углом θ=35°, смаз Δ=14 пкс. На рис. 1, б, представлен результат восстановления изображения путем решения ИУ (1) методом РТ согласно (2) при $\alpha=10^{-4}$ и методом ПФВ согласно (3) при $K=10^{-4}$ при точных параметрах смаза θ=35° и Δ=14 пкс. При этом использованы m-функции системы MatLab fspecial.m, imfilter.m (прямая задача), а также deconvreg.m и deconvwnr.m (обратная

задача) [7]. На рис. 1, в, приведено восстановленное изображение при неточных значениях параметров смаза $\tilde{\theta} = 37°$ и $\tilde{\Delta} = 16$ пкс (при $\alpha = K = 10^{-3}$)[1]. Хотя $\tilde{\theta}$ и $\tilde{\Delta}$ незначительно отличаются от точных θ и Δ, восстановление получилось неудовлетворительным (рис. 1, в).



а б в

Рис. 1. Изображения: смазанное (а); восстановленное с точными параметрами Δ и θ (б); восстановленное с неточными параметрами (в)

На рис. 2, а, представлен аналогичный пример с цветным (rgb) дефокусированным изображением астрономического объекта (галактики) m83.jpg 378×400×3 пкс. В данном примере ФРТ – однородный диск радиуса ρ=10 пкс. На рис. 2, б, представлен результат восстановления изображения путем решения ИУ (1) методами РТ и ПФВ при точном параметре дефокусирования ρ=10 пкс ($\alpha = K = 10^{-4}$). А на рис. 2, в, приведено восстановленное изображение при ошибочном значении параметра дефокусирования $\tilde{\rho} = 11$ пкс ($\alpha = K = 2 \cdot 10^{-3}$). Хотя $\tilde{\rho}$ незначительно отличается от точного ρ, восстановление получилось неудовлетворительным (рис. 2, в)[2].

а б в

Рис. 2. Изображения: дефокусированное (а); восстановленное с точным параметром ρ (б); восстановленное с неточным параметром (в)

Эти и другие примеры [3, 5, 6] указывают на то, что необходимо применить некоторый метод повышения точности оценки параметров искажения изображения, другими словами, повышения точности знания ФРТ. Одним из таких методов является спектральный метод оценки ФРТ [3, 5, 6], в котором параметры ФРТ определяются по Фурье-спектру искаженного изображения.

В данной работе нашел дальнейшее развитие спектральный метод, в частности выполнена его проверка на ряде искаженных изображений.

[1] Значения параметров α и *K* выбраны способом подбора: путем вывода на дисплей изображений w_α и w_K при различных α и *K* и визуального выбора значений α и *K*, дающих наилучшее восстанавление изображения.

[2] Данный пример приведен для сравнения с результатами математического восстановления изображений, полученных космическим телескопом «Хаббл» [4] при наличии остаточной сферической аберрации зеркала телескопа, равносильной дефокусированию.

Отмечены следующие существующие способы оценки параметров искажения изображения: по штрихам на изображении [3], по размытию точек на изображении в случае дефокусирования [2, 3], по Фурье-спектру изображения [1] и т.д. Отмечены активно развиваемые методы «слепой» деконволюции [2, 7] оценки ФРТ, а также работа [8], в которой развит устойчивый алгоритм восстановления изображений при неточно известной аппаратной функции (или ФРТ).

Спектральный метод. Обозначим через $g(x,y)$ распределение интенсивностей по искаженному изображению, причем ось x направим горизонтально, а y – вертикально вниз.

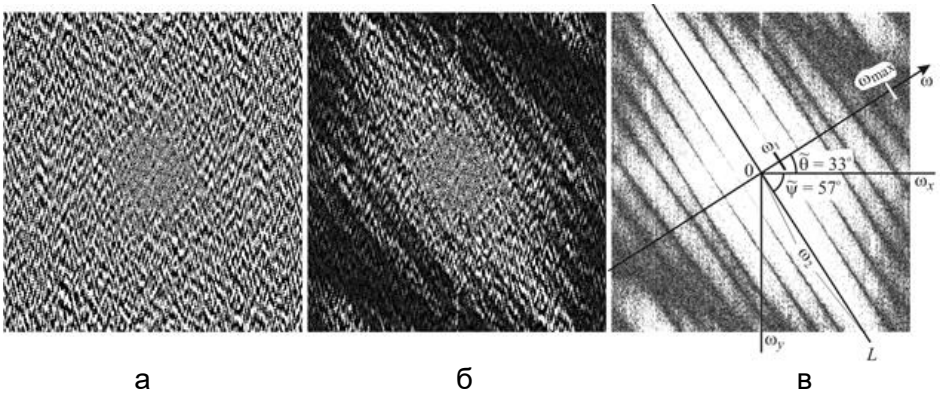
Выполним двумерное преобразование Фурье (ПФ) искаженного изображения $g(x,y)$:

$$G(\omega_x,\omega_y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{i(\omega_x x+\omega_y y)}\,dx\,dy, \qquad (4)$$

где ω_x и ω_y – часто́ты Фурье, направленные горизонтально и вертикально, как x и y. Полагаем, что ПФ (4) вычисляется через ДПФ (дискретное ПФ) с помощью m-функции fft2.m с добавлением центрирования ДПФ [7, С. 126] с помощью fftshift.m. Получим комплексное ДПФ (Фурье-спектр) $G(\omega_x,\omega_y)$, которое выводим в виде $\mathrm{Re}\,G(\omega_x,\omega_y)$ или $|G(\omega_x,\omega_y)|$.

Оценки параметров смаза изображения. Рассмотрим смазанное изображение томограммы-фантома (рис. 1, а). На смазанном изображении введем (мысленно) дополнительные оси: x' вдоль смаза и y' перпендикулярно смазу, а на спектрограмме введем ось ω вдоль смаза. В результате смазывания изображения вдоль x' происходит сжатие Фурье-спектра $G(\omega_x,\omega_y)$ вдоль оси ω (уменьшение ω_{max}), причем тем большее, чем больше смаз Δ. Это связано с тем, что при смазывании подавляются высокие часто́ты Фурье [1].

На рис. 3 представлены три варианта Фурье-спектра (в виде центрированного ДПФ) изображения mrt-1-02.bmp. На рис. 3, а – спектр неискаженного изображения $\mathrm{Re}\,G(\omega_x,\omega_y)$ (для сравнения), на рис. 3, б – вещественная часть спектра смазанного изображения $\mathrm{Re}\,G(\omega_x,\omega_y)$, а на рис. 3, в – спектр по модулю смазанного изображения $|G(\omega_x,\omega_y)|$. Рис. 3 показывает, что вид спектров существенно разный. Это дает возможность определить, смазано ли в принципе изображение. Далее, сравнение рис. 3, б, в, показывает, что модуль спектра $|G(\omega_x,\omega_y)|$ дает более четкую картину, чем $\mathrm{Re}\,G(\omega_x,\omega_y)$.



а б в

Рис. 3. ПФ неискаженного изображения $\mathrm{Re}\,G(\omega_x,\omega_y)$ (а); ПФ смазанного изображения $\mathrm{Re}\,G(\omega_x,\omega_y)$ (б); ПФ смазанного изображения $|G(\omega_x,\omega_y)|$ (в)

Оценка величины смаза изображения. Спектр смазанного изображения (рис. 3, в) состоит из набора почти параллельных линий и центрального квазиэллипса (ср. [1–3, 5, 6]).

Проведем через квазиэллипс среднюю линию L и перпендикулярную ей ось ω, а также горизонтальную ось ω_x и вертикальную ось ω_y. Отметим также на оси ω первый (при $\omega = \omega_1$) и последний (при $\omega = \omega_{\max}$) нули спектра $|G(\omega)|$. Как показано в [6], величина смаза равна

$$\Delta = 2\frac{\omega_{\max}}{\omega_1}. \qquad (5)$$

Применительно к рассматриваемому примеру, безразмерное отношение $\omega_{\max}/\omega_1$ (в любых, но одинаковых единицах: в пикселях, сантиметрах, частотах Найквиста и т.д.) оценено по нескольким замерам на рис. 3, в, как $7{,}02 \pm 0{,}08$ пкс. Поэтому $\Delta = 14{,}04 \pm 0{,}16$ пкс, что близко к точному значению $\Delta = 14$ пкс.

В работе [5] для определения Δ использована формула: $\Delta = 2\omega_2/\omega_1$ (см. рис. 3, в). Однако формула (5) дает более точный результат (см. также [6]).

Оценка направления смаза. По рис. 3, в, определим угол $\tilde{\theta}$ между горизонтальной осью ω_x и осью ω, а также угол $\tilde{\psi} = 90° - \tilde{\theta}$ (измеренные углы). Однако углы $\tilde{\theta}$ и $\tilde{\psi}$, вообще говоря, не совпадают с истинными углами θ и ψ. Это связано с тем, что изображение на рис. 1 и его спектр на рис. 3 являются, вообще говоря, прямоугольными размером $M \times N$ (M строк и N столбцов). Их отношение $r = M/N$. Лишь при $M = N$, т.е. при $r = 1$ (в случае квадратного изображения) $\tilde{\theta} = \theta$ и $\tilde{\psi} = \psi$.

Для определения истинных углов θ и ψ учтем, что $\operatorname{tg}\psi$ равен коэффициенту наклона любой прямой линии, а при растяжении или сжатии изображения, т.е. при изменении r коэффициент наклона изменяется в r раз: $\operatorname{tg}\tilde{\psi} = r \cdot \operatorname{tg}\psi$, откуда истинный угол ψ равен

$$\psi = \operatorname{arctg}\left(\frac{\operatorname{tg}\tilde{\psi}}{r}\right) \qquad (6)$$

и $\theta = 90° - \psi$.

В данном примере по рис. 3 по нескольким замерам определили: $\tilde{\theta} \approx 33°{,}0 \pm 0{,}4$, $\tilde{\psi} = 90° - \tilde{\theta} \approx 57°{,}0 \pm 0{,}6$, $M = 407$, $N = 380$, $r = M/N = 1{,}071$. Тогда согласно (6), получим: $\psi = 55°{,}2 \pm 0{,}6$, а $\theta = 34°{,}8 \pm 0{,}4$, что близко к точному значению угла смаза $\theta = 35°$.

Итак, с помощью спектрального метода были определены с хорошей точностью величина смаза (целое число): $\Delta = 14$ пкс и угол смаза: $\theta = 34°{,}8 \pm 0{,}4 \approx 35°$.

Теперь, используя найденные по спектру значения Δ и θ, возможно получить высококачественное восстановление изображения томограммы-фантома путем решения ИУ (1) методом РТ и параметрической фильтрации Винера (ПФВ) с использованием m-функций deconvreg.m и deconvwnr.m. При этом (ФРТ, PSF) вычислялась с помощью m-функции fspecial.m [7]. Результат восстановления – на рис. 1, б.

Оценка параметра дефокусирования изображения. Рассмотрен простейший вариант дефокусирования, когда каждая точка на объекте превращается на его изображении в однородный круг (диск) радиуса ρ плотности $1/\pi\rho^2$ [5]. Это может иметь место в случае тонкой линзы с круговой апертурой [3, С. 193]. Рассмотрим один такой круг. Двумерное преобразование Фурье однородного круга радиуса ρ (его оптическая передаточная функция – ОПФ) выражается через одномерное преобразование Ганкеля [9, С. 69], [10, С. 249]:

$$F(\omega_1, \omega_2) = F(\omega) = \iint_D e^{i(\omega_1 x + \omega_2 y)}\, dx\, dy = \frac{2\pi}{\pi\rho^2}\int_0^{\rho} J_0(\omega r)\, r\, dr = \frac{2}{\omega^2\rho^2}\int_0^{\omega\rho} J_0(z)\, z\, dz, \qquad (7)$$

где D – область круга; $\omega = \sqrt{\omega_1^2 + \omega_2^2}$, а $J_0(z)$ – функция Бесселя 1-го рода 0-го порядка. Последний интеграл в (7) равен [10, С. 668]

$$\int_0^{\omega\rho} J_0(z)\, z\, dz = \omega\rho\, J_1(\omega\rho)\,, \qquad (8)$$

где $J_1(\omega\rho)$ – функция Бесселя 1-го рода 1-го порядка. Учитывая (8), получим (ср. [1, С. 100]):

$$F(\omega) = \frac{2}{\omega\rho} J_1(\omega\rho)\,.$$

На рис. 4 представлена функция Бесселя $J_1(\omega\rho)$ (ср. [10, С. 669]). Видим, что $J_1(\omega\rho)$ имеет следующие нули:

$$\omega\rho = 0;\ 3{,}84;\ 7{,}02;\ 10{,}16;\ 13{,}32;\ \ldots \qquad (9)$$

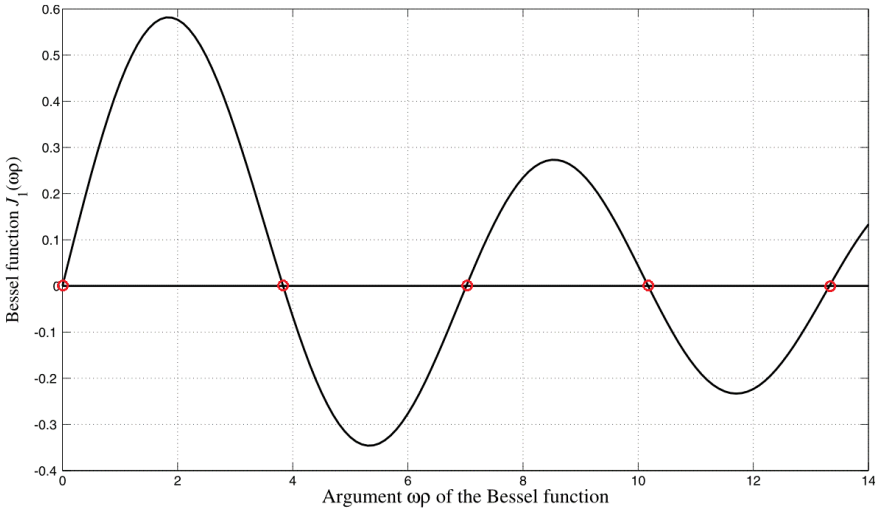
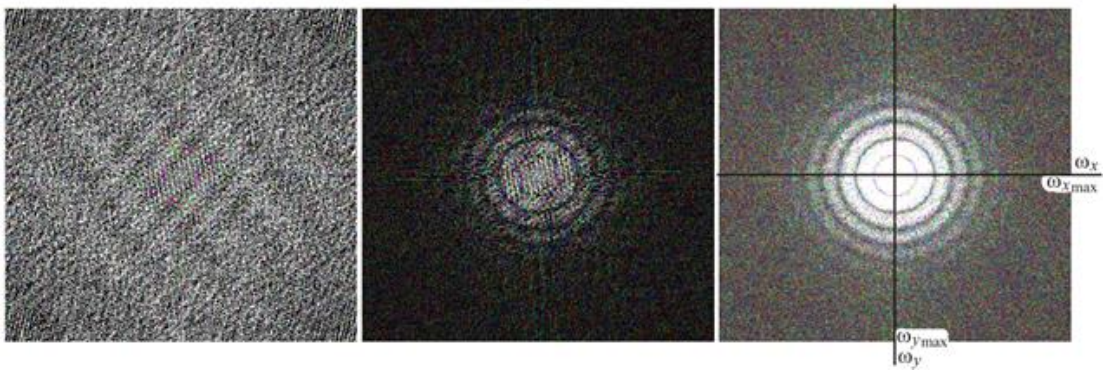


Рис. 4. Функция Бесселя $J_1(\omega\rho)$ 1-го рода 1-го порядка

Эти нули соответствуют черным контурам-эллипсам на рис. 5, б, а также на более контрастном рис. 5, в (модуль спектра) (ср. [3, С. 216]).



а б в

Рис. 5. ПФ $\mathrm{Re}\, G(\omega_x, \omega_y)$ неискаженного изображения m83.jpg (а); ПФ дефокусированного изображения $\mathrm{Re}\, G(\omega_x, \omega_y)$ (б); ПФ дефокусированного изображения $|G(\omega_x, \omega_y)|$ (в)

Из (9) имеем:

$$\rho = 3{,}84/\omega_1;\ 7{,}02/\omega_2;\ 10{,}16/\omega_3; 13{,}32/\omega_4; \ldots,$$

где $\omega_1, \omega_2, \omega_3, \omega_4, \ldots$ – значения частот Найквиста (но не в пкс), соответствующих полуоси каждого эллипса. При дискретизации максимальная частота Найквиста равна $\omega_{\max}=\pi$ как по горизонтальной, так и по вертикальной осям на рис. 5, в. Тогда часто́ты ω_i, $i=1, 2, 3, 4,\ldots$ будут равны $\omega_i = \omega_{i\,\mathrm{rel}}\,\omega_{\max} = \omega_{i\,\mathrm{rel}}\,\pi$, $i = 1, 2, 3, 4, \ldots$, где $\omega_{i\,\mathrm{rel}} = \omega_i / \omega_{\max}$ – безразмерные отношения.

Получили: $\omega_{1\text{rel}} = 0{,}122$, $\omega_1 = 0{,}383$, $\rho = 10{,}02$; $\omega_{2\text{rel}} = 0{,}226$, $\omega_2 = 0{,}710$, $\rho = 9{,}88$ и т.д. В среднем $\rho = 9{,}95 \pm 0{,}07$, что близко к точному значению $\rho = 10$ пкс.

Только теперь, используя найденное по спектру значение $\rho = 9{,}95$ (округленное до $\rho = 10$), получаем возможность восстановить с повышенной точностью изображение галактики M33 путем решения ИУ (1) методом РТ и ПФВ согласно (2) при $\alpha = K = 10^{-4}$ с использованием m-функций deconvreg.m и deconvwnr.m. Результат восстановления – на рис. 2, б. Видно четкое восстановление изображения, и оно обязано тому, что спектральный метод позволил практически точно определить параметр дефокусирования ρ.

Отмечено, что в работах [3, 5, 6] рассмотрен также вариант дефокусирования в случае, когда ФРТ является гауссианой, и использованы оценки параметра σ гауссианы.

Заключение. В работе подтверждена эффективность спектрального метода определения типа искажения (смаз или дефокусирование) изображения и определения параметров искажений. Предложенный спектральный метод может быть применен для повышения разрешающей способности устройств регистрации изображений (цифровых видеокамер, телескопов, микроскопов, томографов и т.д.).

Литература

1. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах. – Новосибирск: Изд-во НГТУ, 2002. – 352 с.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. – 1072 с.
3. Сизиков В.С. Прямые и обратные задачи восстановления изображений, спектроскопии и томографии с MatLab. – СПб.: Лань, 2017. – 412 с.
4. Saïsse M., Rousselet K., Lazarides E. Modeling technique for the Hubble Space Telescope wave-front deformation // Applied Optics. – 1995. – V. 34. – № 13. – P. 2278–2283.
5. Сизиков В.С. Спектральный способ оценки функции рассеяния точки в задаче устранения искажений изображений // Оптический журнал. – 2017. – Т. 84. – № 2. – С. 36–44.
6. Сизиков В.С., Степанов А.В., Меженин А.В., Бурлов Д.И., Экземпляров Р.А. Определение параметров искажений изображений спектральным способом в задаче обработки снимков поверхности Земли, полученных со спутников и самолетов // Оптический журнал. – 2018. – Т. 85. – № 4. – С. 19–27.
7. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. – М.: Техносфера, 2006. – 616 с.
8. Воскобойников Ю.Е. Комбинированный нелинейный алгоритм восстановления контрастных изображений при неточно заданной аппаратной функции // Автометрия. – 2007. – Т. 43. – № 6. – С. 3–16.
9. Брейсуэлл Р. Преобразование Хартли. – М.: Мир, 1990. – 175 с.
10. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. – М.: Наука, 1968. – 720 с.

Кореньков Юрий Дмитриевич
Год рождения: 1990
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: ged.yuko@gmail.com

Логинов Иван Павлович
Год рождения: 1993
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: ivan.p.loginov@gmail.com

Кузенкова Елизавета Владимировна
Год рождения: 1999
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P3217
Направление подготовки: 09.03.04 – Программная инженерия
e-mail: kuzenkova.el@yandex.ru

Лаздин Артур Вячеславович
Год рождения: 1962
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: lazdin@yandex.ru

УДК 004.453

ПРИМЕНЕНИЕ ДЕКЛАРАТИВНОЙ МОДЕЛИ ОПИСАНИЯ ЦЕЛЕВЫХ АРХИТЕКТУР К СУЩЕСТВУЮЩИМ ISA

Кореньков Ю.Д., Логинов И.П., Кузенкова Е.В.
Научный руководитель – к.т.н., доцент Лаздин А.В.

Работа выполнена в рамках темы НИР № 619296 «Разработка методов создания и внедрения киберфизических систем».

Показаны некоторые неочевидные проблемы построения модели целевых архитектур и ее применения для описания существующих аппаратных архитектур. Рассмотрены способы интеграции таких описаний в конвейер компилятора с сохранением их декларативности.
Ключевые слова: компиляторы, методы трансляции, декларативные описания, машинные языки.

На сегодняшний день наиболее распространенным способом создания трансляторов программного кода в машинный является ручное их написание с последующей статической компиляцией такого транслятора. Ввиду сложности предметной области промежуточных

представлений программ и специфичности предметной области программной модели целевого набора инструкций, разработка и поддержка такого рода инструментария требует специфических компетенций для каждого отдельно взятого целевого окружения. Однако алгоритмы компиляции и методов трансляции не являются специфическими для той или иной аппаратуры, их формальные описания применимы к любым конкретным моделям вычислений, обладающим соответствующими аспектами, рассматриваемыми в тех или иных алгоритмах.

Примером таких аспектов могут являться элементы модели вычисления стековых вычислителей: стек и принцип организации инструкций для такого вычислителя. Эти аспекты являются общими для всех стековых вычислителей. Таким образом, возможно создание транслятора программы в код для произвольного стекового вычислителя, но как в таком случае выполнять генерацию кода, ведь машинные коды различных машин могут быть представлены разными битовыми паттернами? – такие наборы битовых паттернов могут быть представлены с помощью декларативных описаний.

В зависимости от модели разработки приложения, декларативные способы описания входных данных, управляющие логикой работы этого приложения, могут иметь значительные преимущества, например, для них, в отличие от императивных алгоритмов трансляции, гораздо проще определить критерии корректности и возможно проверять корректность таких данных методами статической верификации.

Вручную закодированный алгоритм дизассемблирования машинного кода для архитектуры x86 (libdisasm) занимает порядка тысячи строк и включает в себя распознающую логику, написанную специфически с учетом машинных кодов данного набора инструкций. Тестирование такой логики требует явного прогона всех вариантов ветвления распознающего алгоритма и их комбинаций для различных составных машинных кодов. Ручная реализация распознавателя для еще одной архитектуры – повторение всей этой работы. Реализация кодогенератора, отладчика таким методом – также потребует заложить в логику этих инструментов информацию, исходящую из структуры и организации системы команд конкретной машины, несмотря на обобщенность самих алгоритмов инструментария и принципов организации различных машин.

Несмотря на использование методов обобщенного программирования при создании таких широко известных инструментариев, как GCC и LLVM, они, в конечном счете, также не используют обобщенных методов ассемблирования-дизассемблирования, что наглядно показывает диаграмма функциональности LLVM (рис. 1) [1].

	Target										
Feature	**ARM**	**Hexagon**	**MSP430**	**Mips**	**NVPTX**	**PowerPC**	**Sparc**	**SystemZ**	**X86**	**XCore**	**eBPF**
is generally reliable	Y	Y	?	Y	Y	Y	Y	Y	Y	Y	Y
assembly parser	N	N	N	N	N	N	N	Y	Y	N	N
disassembler	Y	N	N	N	N/A	N	Y	N	Y	Y	Y
inline asm	Y	Y	?	N	Y	Y	?	Y	Y	Y	N
jit	*	N	?	Y	N/A	Y	?	Y	Y	N	Y
.o file writing	N	N	N	N	N/A	N	N	Y	Y	N	Y
tail calls	Y	Y	?	N	N	Y	?	N	Y	N	N
segmented stacks	N	N	N	N	N	N	N	N	*	N	N

Рис. 1. Матрица функциональных возможностей LLVM [1]

Описания архитектур этих инструментов, используемые статически, включают только аспекты модели вычислений, но не включают способ кодирования системы команд и связи абстрактных команд с вариантами операционных кодов. По этой причине для них отсутствуют

методы верификации корректности, такие как проверка отсутствия взаимоисключающих инструкций.

Разработана нотация для записи декларативных определений наборов инструкций целевых платформ, предназначенная восполнить вышеобозначенные недостатки широко распространенных подходов [2].

Эта нотация позволяет обобщенно формулировать варианты моделей вычисления, произвольным образом комбинирующих элементы стековых, регистровых и RAM-машин, сопутствующие им наборы инструкций и форматы мнемоник. Такого набора информации достаточно для выполнения трансляции кода процедур в поток машинных инструкций. Затем нотация была расширена для включения в нее возможности описания абстракции состояния методов, что позволяет не просто транслировать логику, но выполнять трансляцию сложных программ, состоящих из многих процедур, в соответствии с конвенциями той или иной целевой платформы, не имея для нее специфической логики трансляции. Для такого обобщенного транслятора, помимо промежуточного представления самой программы, описание целевой архитектуры является одним из входных параметров.

При разработке модели этих описаний авторы столкнулись с вариативностью внутренних структур состояний реальных вычислительных моделей, которые было необходимо сформулировать единообразно. Примерами таких конфликтов являются различия в способах организации локального состояния процедур, локальных переменных, различия в способах организации конвенций вызовов, и т.д.

Так, например, информация об организации аппаратного стека для его поддержки транслятором и отладчиком потребовала введения новых аспектов в первичную версию нотации. А поддержка генерации процедур в соответствии с конвенциями была организована с помощью распространенного метода интринсиков – предопределенных последовательностей команд, выполняющих заданные моделью функции в рамках модели программы.

В компиляторах Intel C/C++ такими интринсиками в язык Си введены многие инструкции из наборов SSE и других, например, инструкция lzcnt имеет следующие прототипы фукнкций (Intel Software Developer Manuals):

Intel C/C++ Compiler Intrinsic Equivalent
LZCNT:
  unsigned __int32 _lzcnt_u32(unsigned __int32 src);
LZCNT:
        unsigned __int64 _lzcnt_u64(unsigned __int64 src);

Использование таких функций в теле программы на языке Си приводит к их трансляции непосредственно в соответствующие машинные коды исходных инструкций.

В рамках авторской нотации подобным образом организуется возможность определения конвенций вызовов: с помощью интринсиков префикса и постфикса функции. Приведем пример определения эпилога функции:

```
architecture x86 {
 // ...
 instructions:
 // должны быть определены кодировки необходимых для эпилога инструкций
 instruction pop1 = { 1000 1111, 11 000 reg as to };
 instruction ret-near-1 = { 11000011 };

 mnemonics:
 // и их мнемоники
 mnemonic pop for pop1(to) "{1}";
 mnemonic ret for ret-near-1();
}
```

После того как машинные коды и мнемоники для необходимых в составе эпилога инструкций определены, они могут быть использованы для описания самого эпилога:

```
environment x86-test {
 // после этого
 convention myConv {
 // ...
 epilog {
 pop esp
 ret
 }
 }
}
```

Имея на входе такие определения, компилятор может использовать их для оформления функций и процедур при генерации машинного кода. Меняя эти определения, получена возможность генерировать код, соответствующий различным конвенциям вызовов, операционным системам, аппаратным моделям вычислительных машин.

В противовес распознавателю инструкций, упомянутому выше, – порядка тысячи строк для распознавания всех инструкций только архитектуры x86, – обобщенная реализация распознавателя любых инструкций любых машин на основе таких определений занимает около 150 строк.

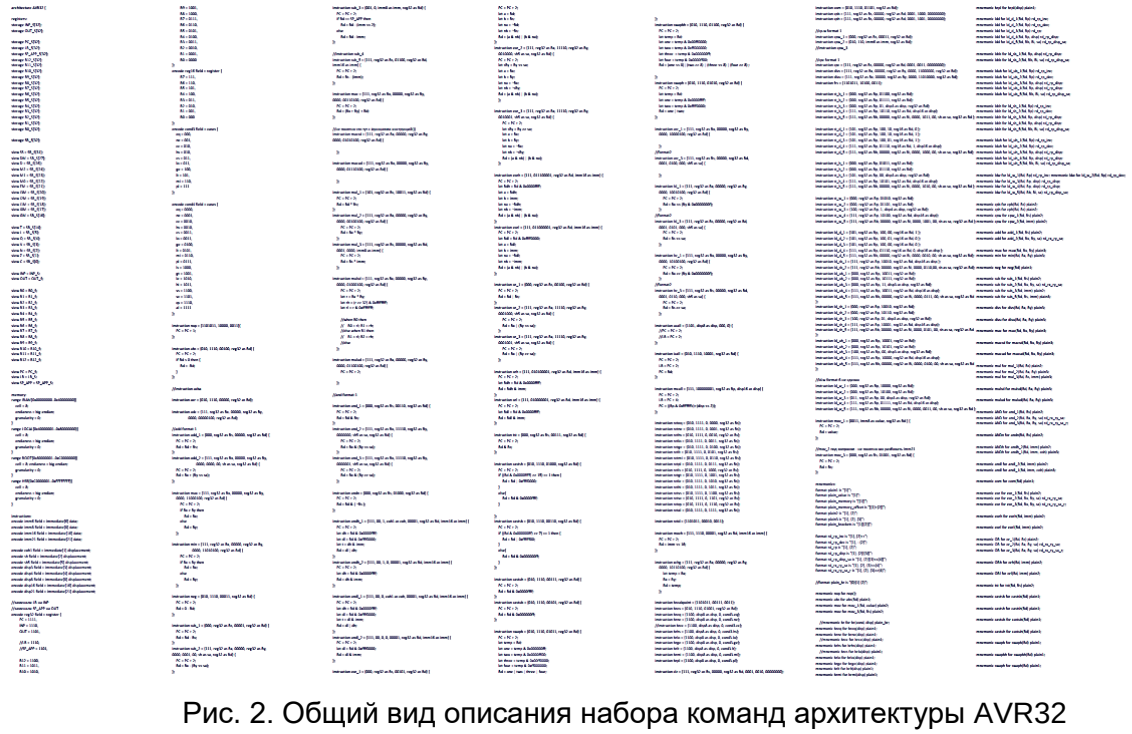


Рис. 2. Общий вид описания набора команд архитектуры AVR32

Ключевые расширения аспектов созданной авторами нотации в ходе применения ее для трансляции кода реальных вычислительных машин включают в себя преимущественно аспекты программной модели – способов организации конвенций вызовов, локальных состояний программы, сложных паттернов передачи управления и тому подобное. Они не являются обязательными для трансляции кода как таковой, но позволяют ограничить ее общепринятыми принципами выполняемых модулей, содержащих транслированный код программ.

В дополнение к уже введенным аспектам, возможно расширение нотации аспектами, связанными с параллелизмом инструкций и временами их выполнения при различных сочетаниях в ходе конвейерного исполнения. Такая информация позволяет реализовать в

потенциальном обобщенном компиляторе не только обобщенную трансляцию кода между различными моделями вычислений, но и обобщенную оптимизацию по времени.

Кроме прочих, созданная нотация была применена для описания набора инструкций архитектуры AVR32 (рис. 2). На его основе осуществима не только трансляция программ, но и их дизассемблирование, восстановление информации о структуре алгоритма с дальнейшим анализом, отладка уже скомпилированных программ с анализом их состояния и поддержка соответствующих возможностей для интегрированных сред разработки (IDE). При этом не требуется дублирование информации о наборе команд и модели вычисления в коде различных инструментов, предоставляющих конечному пользователю упомянутую функциональность [3–7].

Литература

1. LLVM [Электронный ресурс]. – Режим доступа: http://llvm.org/docs/CodeGenerator.html#target-feature-matrix (дата обращения: 04.01.2019).
2. Korenkov I., Loginov I., Dergachev A., Lazdin A. Declarative target architecture definition for data-driven development toolchain // 18th International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management. – 2018. – V. 18. – № 2.1. – P. 271–278.
3. Novillo D. GCC an architectural overview, current status, and future directions // Proceedings of the Linux Symposium. – 2006. – V. 2. – P. 185–200.
4. Lattner C., Adve V. Architecture for a next-generation GCC [Электронный ресурс]. – Режим доступа: https://llvm.org/pubs/2003-05-01-GCCSummit2003.pdf (дата обращения: 04.01.2019).
5. Lattner C., Adve V. LLVM: A compilation framework for lifelong program analysis & transformation // Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization. – 2004. – P. 75.
6. Pandey M., Sarda S. LLVM cookbook. – Packt Publishing Ltd, 2015. – 296 p.
7. Korenkov Y., Loginov I., Lazdin A. PEG-based language workbench // Open Innovations Association, 2015 17th Conference of. – 2015. – P. 75–81.

Кудрявых Антон Юрьевич
Год рождения: 1994
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4255
Направление подготовки: 09.04.01 – Информатика и вычислительная техника
e-mail: dart-shaul@mail.ru

Авксентьева Елена Юрьевна
Год рождения: 1981
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: avksentievaelena@rambler.ru

УДК 004.41

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИСПОЛЬЗОВАНИЯ МОНОЛИТНОЙ И МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ ПРИЛОЖЕНИЙ

Кудрявых А.Ю.
Научный руководитель – к.педагог.н., доцент Авксентьева Е.Ю.

Работа выполнена в рамках темы НИР № 615892 «Исследование и разработка в области информационных технологий».

В работе были исследованы особенности построения архитектуры сложных web-приложений и сервисов, рассмотрены два основных противоположных по своей сути архитектурных решения. Были выявлены их отличительные особенности, проанализированы преимущества и недостатки, а также приведены рекомендации по выбору и внедрению того или иного варианта. Целью данной работы было определение границ использования этих различных подходов в организации web-систем с учетом их особенностей и возможностей.
Ключевые слова: web, монолитная архитектура, микросервисная архитектура, web-системы, принципы построения, разработка, внедрение.

Введение. Изначально, на стадии становления и роста глобальной сети, web-системы не отличались большой сложностью в своем создании и проектировании, они были относительно просты и писались для удовлетворения небольших локальных задач. Однако впоследствии появление новых технологий в разработке и проектировании, а также возрастающая популяризация глобальной сети, привели к существенному усложнению архитектуры более современных web-приложений [1–4].

Тем не менее, долгое время существовала практика инкапсуляции практически всех компонентов системы в единую тесно связанную монолитную структуру. Это архитектура и была в последствии названа монолитной, поскольку включала в себя все компоненты сложной системы в едином узле и работала, зачастую, на одном физическом сервере, почти не имея внешних связей. За счет новшеств, которые привнесли современные языки разработки (такие как объектно-ориентированный подход в программировании, разбиение программ на подмодули, выделение API этих самых модулей), данный подход успешно применялся и использовался в крупных информационных системах даже несмотря на значительно

возрастающую сложность, держать все модули и клубок из их взаимосвязей под одним «капотом».

Способствовали этому и все улучшающиеся инструменты в разработке. Появлялись новые библиотеки, фреймворки, улучшались сами языки и технологии, позволяющие разрабатывать и поддерживать web-сервисы. Однако с возросшей сложностью пришла и высокая нагрузка на эти системы, стали появляться проблемы с масштабируемостью, распределенностью и с особенностью организации баз данных в таких проектах.

Решением возникших проблем и стала так называемая микросервисная архитектура, т.е. переход от централизованной сборки всего web-приложения в единую сущность к разбиению этого самого приложения на отдельные полностью обособленные друг от друга модули, сервисы, функционирующие изолированно на своих собственных платформах.

Однако сама концепция микросервисов не является панацеей и несет в себе много недостатков, о которых следует помнить инженеру при организации архитектуры приложения.

Микросервисная архитектура. Микросервисы в текущем их представлении являются самой востребованной архитектурной парадигмой, особенно в сложных системах. Огромное количество разработчиков применяют их в построении новых и даже в доработке старых web-приложений и сервисов. В отдельных случаях происходят даже попытки растащить сложное монолитное приложение на отдельные микросервисы, что является достаточно трудоемким процессом.

Популярность и любовь разработчиков к такому построению приложений пришла неспроста. Сама концепция дефрагментации сложных систем на более простые лежала на поверхности и применялась даже в современных языках объектно-ориентированного программирования, однако отсутствие соответствующего инструментария и, что самое главное, необходимости в таком решении до поры до времени тормозило данный процесс.

В первую очередь микросервисная архитектура хороша своей абстракцией небольших компонентов систем друг от друга. Разделение разрабатываемых модулей позволяет разработчикам использовать те инструменты, подходы и практики, которые необходимы в данной конкретной ситуации. Монолитная архитектура привязывала всю систему к определенному стеку технологий, так как вся система должна была функционировать в тесной связке друг с другом. Микросервисы же развязывают руки в применении тех или иных технологий. К примеру, компонент голосового помощника с применением нейронных сетей, удобней писать с использованием Python и соответствующего ему инструментария, а сложную серверную прослойку писать уже на Java. Другой микросервис, полностью основанный на ассинхронных вызовах, удобней написать на NodeJS. Кроме того, можно использовать разные базы данных для каждого решения, разные библиотеки, инструменты и прочее, которые не могли бы функционировать в рамках одного проекта.

Во вторую очередь – это в разы возрастающая масштабируемость системы. Вместо того чтобы масштабировать всю систему целиком, можно масштабировать отдельные компоненты системы, причем масштабировать как вертикально, так и горизонтально. За счет этого не возникает проблем с дублированием других сервисов системы, которым масштабирование и не требуется.

В третью очередь – это независимость развертывания. Каждый микросервис представляет собой упрощенную систему со своей окружающей средой, а значит, и развертывать ее значительно проще. При этом при развертывании можно не привязывать к одной конкретной технологии, серверу или фреймворку, а использовать именно то, что лучше будет работать в рамках разработанного микросервиса.

И наконец, в-четвертых – микросервисная архитектура предоставляет широкий плацдарм для улучшений и модернизации всей системы. Если со временем система растет и развивается, перерастая в нечто большее, гораздо удобней ее дорабатывать с использованием

локальных сервисов, чем перерабатывать огромное монолитное приложение. Именно по этой причине, зачастую, очень многие и начинают «растаскивать» свои web-системы на отдельные компоненты, дабы была возможность безопасно и малой кровью внедрить новые компоненты в уже существующую систему.

Проблемы микросервисной архитектуры и плюсы монолитной. Однако, несмотря на широкую популярность микросервисов, огромное количество проектов продолжает разрабатываться с использованием монолитной архитектуры приложения. Весомые плюсы, которые дают микросервисы, скрывают и не менее весомые минусы, о которых разработчики порой забывают.

В первую очередь речь идет о сложности эксплуатации. Поддерживать множество сервисов, пусть и небольших, порой экономически менее целесообразно, чем один единственный монолит. Да, его сложно развернуть, с ним есть проблемы при обновлении, патчах, новых версиях и т.д., но все же администраторам приходится обслуживать лишь одну базу-две базы данных и несколько серверов. Конечно, современные средства виртуализации и контейнеры частично решают и эту проблему, однако всех сложностей в эксплуатации это не снимает.

Вторая проблема, которую несет в себе микросервисная архитектура – это нюансы, связанные с распределенностью транзакций. В ситуациях, когда необходима транзакционная целость, микросервисы могут оказаться не лучшим решением. Помимо этого, возрастает и время отклика всей системы. Каждой транзакции необходимо пройти через множество обособленных друг от друга узлов, что увеличивает время обработки по сравнению с обработкой транзакции в одной системе.

Третья проблема микросервисов – сложности в организации и время разработки. Поскольку каждый отдельный компонент системы существует обособлено друг от друга, ему необходимо продумать грамотный интерфейс взаимодействия (API). Любые ошибки на данной стадии проектирования влекут за собой огромные сложности по их устранению, так как данные в рамках таких систем ходят по целым цепочкам микросервисов. То, что в монолите устраняется относительно легко, в микросервисах несет в себе дополнительные сложности в переписывании и модернизации интерфейсов взаимодействия, причем с большой долей вероятности сразу нескольких микросервисов. Помимо этого, так как каждый сервис существует на своем отдельном узле (сервере), то разработчику помимо реализации API также нужно разрабатывать и многие шаблонные вещи для развертывания и функционирования. В монолите код обращается непосредственно к коду, и нет смысла реализовывать контроллеры и задумываться над особенностями развертки в той или иной серверной среде.

Помимо этого, у микросервисов существует проблемы и с версионностью, когда при длительной разработке иногда можно оказаться в ситуации, что не совсем понятно какие из версий сервисов корректно функционируют друг с другом. И проблемы в размытости границ и сложности управления, когда команде не до конца понятны уровни и границы разрабатываемого приложения. Это ведет и к дублированию кода (к примеру, дублирование проверок входных данных) и функционала, и к проблемам, связанным со связью всей системы.

Разумеется, недостатки микросервисов – это преимущества монолитов и наоборот. Там, где одна из архитектурных концепций выигрывает, начинает проигрывать другая. Монолит, хорош именно своей монолитностью, инкапсуляцией всей сложной системы в одну сущность, но в то же самое время эта инкапсуляция и есть его недостаток.

Критерии выбора архитектурной платформы. Важно понимать, что выбор между тем или иным подходом в организации приложения необычайно важен для всего проекта в целом. Неправильная расстановка требований и приоритетов, неправильное выделение границ подсистем, даже неправильная оценка дальнейшего развития может негативно сказаться на всей разработке, а в худшем случае и вообще ее похоронить.

При выборе той или иной архитектуры важно понимать в первую очередь весь объем проекта с учетом его дальнейшей модернизации и развития. Если web-сервис является небольшим и компактным – стоит обратить внимание на монолит. Распределять небольшие приложения на микросервисы имеет смысл только лишь при наличии сложной масштабируемости отдельных ее компонентов. И напротив, если планируется разработка сложной системы, отдельные компоненты которой сами по себе вполне способны сойти за свою собственную систему – лучше воспользоваться преимуществами, которые дают микросервисы.

Не стоит забывать и о размерах команды по разработке, их распределенности и количестве человек. Если можно усадить всю команду по разработке за один и стол и за счет коммуникации обойти все острые углы в проекте, то можно отказаться от микросервисов в пользу монолитного web-приложения. В том же случае, если команд несколько, или команда большая, а каждая команда или человек трудится над своим рядом задач, имеет смысл разбивать приложение. Однако при таком подходе необходимы люди, которые будут видеть всю систему целиком, продумывать средства коммуникации сервисов друг с другом, но при этом не будут глубоко погружаться в тонкости каждой подсистемы. При этом необходимы и компетентные DevOps инженеры, разбирающиеся в оркестровке, контейнеризации и внедрению, способные поддерживать проект.

Заключение. В работе проанализированы два основных подхода проектирования сложных web-систем и приложений. Были детально описаны их отличительные особенности, преимущества и недостатки, а также методы их использования в проектах различных категорий.

Однако сделать окончательный вывод в пользу того или иного подхода нельзя. К сожалению, на текущий момент в современном мире нет панацеи или более-менее обобщенной архитектуры, подходящей для разного рода задач. Каждая архитектурная платформа из представленных несет свои преимущества и недостатки, которые обязательно необходимо учитывать при выборе. Нельзя без конкретной ситуации сказать, что микросервисы лучше монолитного построения web-приложения, или утверждать обратное – очень многое зависит от проекта, команды, будущего развития системы, ее сложности, нагруженности, распределенности и так далее.

Однако современная практика показывает, что очень много разработчиков просто пытаются следовать трендам не до конца разбираясь, подходят ли они в их конкретной ситуации. Быть на волне новых течений и не стагнировать на старых технологиях в IT-индустрии очень важно, но следует также понимать и тот факт, что не всякая новая технология или подход применим в тот или иной проект.

Литература

1. Ньюмен С. Создание микросервисов. – СПб.: Питер, 2016. – 304 с.
2. Хорсдал К. Микросервисы на платформе .NET. – СПб.: Питер, 2018. – 352 с.
3. Микросервисы: пожалуйста, не нужно! [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/311208/ (дата обращения: 19.01.2019).
4. Микросервисы или монолит: в поисках золотой середины [Электронный ресурс]. – Режим доступа: https://blog.gramant.ru/2017/05/17/microservices-vs-monolith-compromise/ (дата обращения: 16.01.2019).

Кудряшов Артем Петрович
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P4270
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: artk1111@gmail.com

Меженин Александр Владимирович
Год рождения: 1959
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: mejenin@mail.ru

УДК 004.94

МЕТОДЫ ОПТИМИЗАЦИИ СЦЕН В ИГРОВЫХ КОМПЬЮТЕРНЫХ ПРИЛОЖЕНИЯХ

Кудряшов А.П., Меженин А.В.
Научный руководитель – к.т.н., доцент Меженин А.В.

В работе рассмотрены существующие методы оптимизации трехмерных полигональных моделей для приложений реального времени, а также методы оптимизации трехмерных сцен в системе Unreal Engine 4.
Ключевые слова: оптимизация трехмерных моделей, оптимизация трехмерных сцен.

Индустрия игровых приложений, появившаяся в конце прошлого тысячелетия, сейчас активно развивается. Интерес к данной сфере растет с каждым годом, появляется все больше проектов, также растет количество пользователей игровых приложений. Появились различные направления развития индустрии: сегодня проходят глобальные международные чемпионаты по киберспорту, который с 2016 года стал официальным видом спорта в России. Также на базе игровых компьютерных приложений создаются обучающие программы для врачей, военных, детей и людей с ограниченными возможностями. Набирающая популярность и быстро развивающаяся технология дополненной реальности также расширяет применение виртуальных сред, позволяя симулировать различные ситуации, с которыми может столкнуться профессионал в своей дальнейшей работе.

Технический прогресс не стоит на месте и с каждым годом мощность смартфонов, планшетов, игровых приставок и персональных компьютеров повышается. Из-за чего появляются новые возможности по улучшению графики, что в большинстве случаев означает повышение реалистичности изображения. Повышается детализация объектов, улучшается качество освещения и наложения теней, повышаются разрешения текстур, физические свойства объектов становятся более реалистичными. Из-за такого стремительного развития технологий пользователям приходится либо снижать настройки качества графики, либо покупать новые устройства и компоненты. Это может быть выгодно производителям компьютерной техники, но для разработчиков игр это потеря потенциальной аудитории. Возникает потребность в снижении требований к производительности техники без потери качества графики. Для решения такой задачи существуют различные методы оптимизации

игровых компьютерных приложений. Каждый метод эффективен для специфичных ситуаций процесса разработки.

Основой оптимизации игровых компьютерных приложений является снижение количества полигонов в объектах. Таким образом, снижается нагрузка на графический процессор и память используемого устройства, из-за чего повышается количество кадров в секунду. Однако снизить количество полигонов в модели нужно таким образом, чтобы она не сильно потеряла детализацию [1]. Кроме того, оптимизация происходит на этапе рендера в реальном времени, когда используются различные технологии, внедренные в игровой движок.

Оптимизация модели начинается еще на этапе продумывания, когда разработчик определяет характерные детали объекта, его силуэт, который очень важен в компьютерной графике, а также места, где детализацию можно передать не с помощью геометрии, а с помощью текстур. Также при создании модели нужно учитывать, что полигональная сетка объекта должна обладать правильной топологией, которая подразумевает под собой:

- отсутствие треугольных полигонов, способных выдавать артефакты при сглаживании;
- осторожное использование булевых операций и ручная корректировка сетки после их выполнений;
- отсутствие совпадающих, лежащих в одной плоскости полигонов;
- объединение точек, находящихся в одних и тех же координатах;
- стремление к замкнутым контурам.

Также оптимизация затрагивает процесс создания текстур для игровых объектов. Использование различных карт позволяет придать объекту различные характеристики рельефности, отражаемости, прозрачности, что придает объекту дополнительную детализацию. Используются различные методы сжатия текстур. Все это позволяет снизить требования к устройствам, на которых воспроизводится игра или приложение, при этом не теряя в качестве графики.

- Метод низкополигонального моделирования. При создании модели, которая должна быть нацелена на быструю загрузку и использование минимального количество ресурсов персонального компьютера или мобильного устройства, основным методом будет метод низкполигонального моделирования. Метод подразумевает под собой моделирование минимально возможным количеством полигонов так, чтобы их хватало на поддержание формы объекта. Метод не требует наличия высокополигональной модели. Обычно объекты, созданные с использованием этого метода, текстурируются не уникальными текстурами, получаемыми за счет запекания, а текстурами, которые используются множеством других объектов. Данный метод подходит для большинства случаев моделирования для игровых приложений.
- Метод редукции полигонов. Одним из популярных методов оптимизации трехмерных моделей является метод редукции полигонов. Суть метода заключается в сокращении количества полигонов в уже существующей модели путем замены нескольких близких друг к другу полигонов одним, схожим по форме. В большинстве программ для моделирования есть подобная функция, параметры которой обычно дают возможность указать степень редукции, а также другие параметры, позволяющие получить максимально удовлетворяющий требованиям результат. В одной из самых популярных программ для создания трехмерной графики – Autodesk Maya – присутствует команда Reduce, которую можно вызвать из вкладки Polygons, в меню Mesh, далее Reduce [2]. Однако не все модели могут быть успешно оптимизированы автоматическим способом. Тогда используется метод ретопологии [3] (рис. 1, а). Суть метода заключается в ручном создании новой полигональной сетки с меньшим количеством полигонов, поверх старой высокополигональной сетки (рис. 1, б). Так как метод позволяет контролировать степень проработки деталей и степень редукции отдельно в каждом случае, он идеально подходит

для создания объектов со сложной формой и неравномерной сеткой, например, его часто используют для оптимизации моделей персонажей.

а б

Рис. 1. Пример ретопологии высокополигональной модели (а) в низкополигональную (б)

– Метод удаления невидимых частей объекта. Данный метод является основополагающим методом оптимизации модели на этапе моделирования, как и метод низкополигонального моделирования. Суть метода заключается в удалении граней объекта, которые никогда не будут видны пользователю. Для этого моделлеру необходимо знать, где будет располагаться объект в сцене, его окружение, а также со скольких углов и как близко пользователь потенциально будет видеть этот объект. Несмотря на свою простоту и логичность, моделирование с использованием этого метода позволяет сократить количество полигонов до 50%.
– Оптимизация текстур. Все вышеописанные методы непосредственно связаны с топологией модели. Следующим этапом создания трехмерной модели является повышение детализации с помощью текстур. Важной частью оптимизации текстур является создание текстурных атласов [4] (рис. 2). Текстурные атласы – изображения, содержащие в себе комбинацию текстур разных частей объекта и даже разных объектов. Совмещая несколько текстур в одну, а также используя одни текстуры на разные объекты можно снизить нагрузку на персональный компьютер или мобильное устройство.

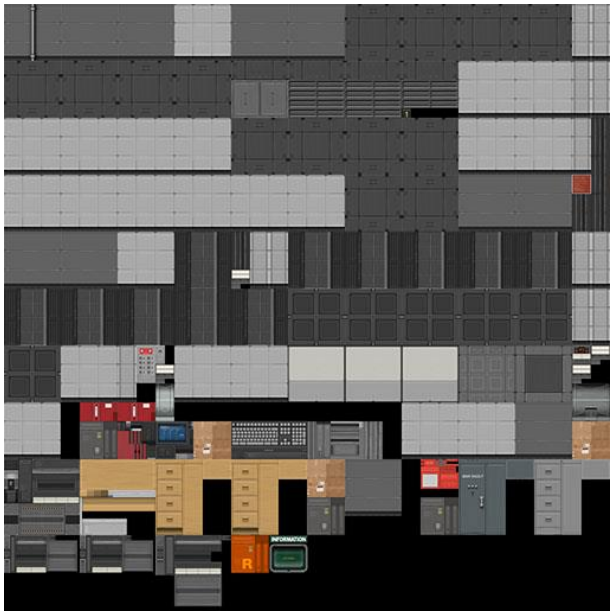


Рис. 2. Текстурный атлас

Непосредственно связан с этой частью моделирования метод запекания текстурных карт [5]. Этот метод подразумевает под собой сохранение визуальной детализации объекта при меньшем количестве полигонов путем переноса информации о рельефе с высокополигональной модели на ее низкополигональную копию, созданную либо методом ретопологии, либо методом редукции полигонов. Также возможно оптимизировать сам файл текстурного изображения. Например, можно использовать каждый канал RGB для отдельной монохромной карты [6].

– Методы оптимизации трехмерной сцены в реальном времени. Рассмотренные ранее методы являются методами, используемыми для оптимизации моделей еще до экспорта в виртуальную игровую среду – игровой движок. Далее будут рассмотрены методы, позволяющие снизить нагрузку на персональный компьютер или мобильные устройства, используя технологии, интегрированные в игровой движок. В качестве примера игрового движка будет Unreal Engine 4. Это современный игровой движок, позволяющий создавать игры и приложения на различных платформах: персональных компьютерах, игровых консолях, а также мобильных устройствах на базе iOS и Android.
– Distance culling. Данный алгоритм позволяет не рендерить, а значит, и не нагружать просчетом графический процессор, те объекты, которые находятся на заданном расстоянии до камеры или, основываясь на том, какой объем экрана занимает объект. Основными особенностями данного метода является то, что алгоритм использует процессор, а не видеокарту, что позволяет сократить число запросов к видеокарте. Также важно заметить, что алгоритм позволяет скрывать только статичную геометрию.
– View Frustum Culling (рис. 3). Данный алгоритм основывается на угле обзора камеры. Все объекты, которые находятся вне зоны видимости камеры, не будут отрендерены. Алгоритм хорошо подходит уровням с большим открытым пространством.

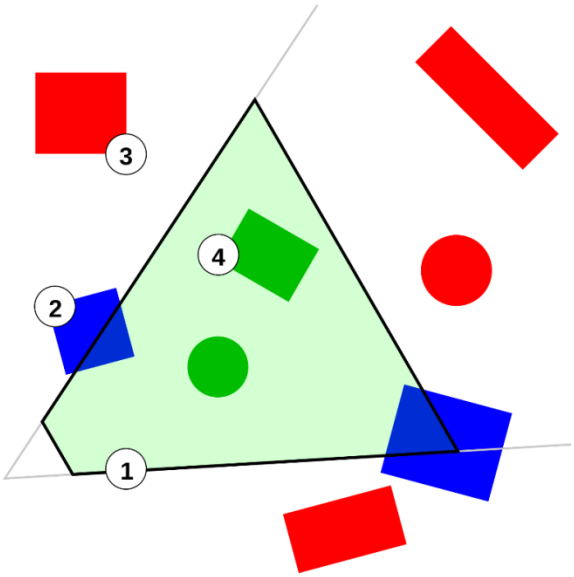


Рис. 3. Схема работы алгоритма View Frustum Culling

Данный алгоритм в игровом движке Unreal Engine 4 работает постоянно, по умолчанию. Также метод не нуждается, почти ни в какой настройке. На рис. 3 видно, что красные объекты отрендерены не будут, так как находятся вне угла обзора камеры, либо скрыты за другими объектами. В то время как зеленые и синии объекты рендереться будут, несмотря на то, что синие объекты не полностью находятся в зоне видимости.

– Precomputed Visibility Culling. Данный метод позволяет рассчитать видимость объектов из всевозможных точек местонахождения камеры заранее, а не просчитывать в режиме реального времени, как это делают другие алгоритмы. Сбор информации о видимости происходит на этапе построения lighting map карты. Данный метод идеально подходит для

маленьких и средних уровней, а также отлично работает с мобильными устройствами, так как в отличие от View Frustum Culling работает не с GPU, а с CPU.

LOD (Levels Of Detail – уровни детализации) – прием в программировании трехмерной графики, заключающийся в создании нескольких вариантов одного объекта с различными степенями детализации, которые переключаются в зависимости от удаления объекта от виртуальной камеры [7]. Другой метод заключается в использовании одной основной, «грубо приближенной», модели и нескольких внешних надстроек к ней. Каждая последующая надстройка к основной модели дополняется элементами детализации пропорционально номеру надстройки, т.е. на самом большом расстоянии будет отображаться единственная главная модель объекта. С приближением же последнего к камере игрока, к конвейеру отрисовки будут последовательно подключаться последующие надстройки деталей. Смысл приема заключается в том, что отображать высокодетализированные объекты, находящиеся на большом расстоянии от виртуальной камеры, нецелесообразно, затратно по вычислительным ресурсам. Использование LOD способно существенно снизить требования к ресурсам компьютера при выводе графики на экран, будь то рендеринг или вывод в реальном времени. С использованием LOD, впрочем, связан широко распространенный баг, при котором на ближних планах отображается низкополигональная модель объекта. Существуют два подхода к управлению детализацией:

1. статический LOD – уровни создаются на этапе препроцессинга и имеют фиксированное разрешение;
2. динамический LOD – создание структуры данных, из которой геометрия с нужной детализацией может быть извлечена в реальном времени во время визуализации.

Все перечисленные методы могут быть использованы для оптимизации сцены в том или ином случае, в зависимости от ситуации. Следует правильно подбирать нужный в данный момент разработки метод для повышения скорости и общего КПД работы.

Литература

1. Особенности создания 3d моделей для игровых движков [Электронный ресурс]. – Режим доступа: http://3d-modeli.net/uroki-3d/6041-osobennostisozdaniya-3dmodeley-dlya-igrovyh-dvizhkov.html (дата обращения: 04.01.2019).
2. Reduce the number of faces in a mesh [Электронный ресурс]. – Режим доступа: https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/Maya-Modeling/files/GUID-227DC277-EB08-4038-932E-ED8B1BC73A0E-htm.html (дата обращения: 04.01.2019).
3. Retopology [Электронный ресурс]. – Режим доступа: https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/Maya-Modeling/files/GUID-74867E07-5CCD-4D55-B60B-C90C3AD65DF4-htm.html (дата обращения: 04.01.2019).
4. Using a Texture Atlas to Optimize Your Game [Электронный ресурс]. – Режим доступа: https://gamedevelopment.tutsplus.com/articles/using-texture-atlas-in-order-to-optimize-your-game--cms-26783 (дата обращения: 04.01.2019).
5. Texture Baking [Электронный ресурс]. – Режим доступа: http://wiki.polycount.com/wiki/Texture_Baking (дата обращения: 04.01.2019).
6. Channel Packing [Электронный ресурс]. – Режим доступа: http://wiki.polycount.com/wiki/ChannelPacking (дата обращения: 04.01.2019).
7. Setting Up Automatic LOD Generation [Электронный ресурс]. – Режим доступа: https://docs.unrealengine.com/en-us/Engine/Content/Types/StaticMeshes/HowTo/AutomaticLODGeneration (дата обращения: 04.01.2019).

Кузенкова Елизавета Владимировна
Год рождения: 1999
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P3217
Направление подготовки: 09.03.04 – Программная инженерия
e-mail: kuzenkova.el@yandex.ru

Кореньков Юрий Дмитриевич
Год рождения: 1990
Университет ИТМО, факультет программной инженерии
и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: ged.yuko@gmail.com

Логинов Иван Павлович
Год рождения: 1993
Университет ИТМО, факультет программной инженерии
и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: ivan.p.loginov@gmail.com

УДК 004.453

СРАВНЕНИЕ СБОРЩИКОВ МУСОРА В РАЗЛИЧНЫХ ПЛАТФОРМАХ

Кузенкова Е.В., Кореньков Ю.Д., Логинов И.П.
Научный руководитель – аспирант Кореньков Ю.Д.

Работа выполнена в рамках темы НИР № 619296 «Разработка методов создания и внедрения киберфизических систем».

Предметом данной работы являлось сравнение деталей реализации сборки мусора для систем управления памятью в различных программных окружениях. Были сформулированы рекомендации для начального выбора платформы при разработке программных систем с большой нагрузкой по динамическому распределению памяти.
Ключевые слова: автоматическое управление памятью, сборщики мусора, алгоритмы сборки мусора, управляемая куча, компиляторы.

Существуют два основных класса методов управления памятью: ручное и автоматическое. Ручное управление памятью заключается в явном кодировании по ходу программы мест, в которых необходимо освободить используемые структуры данных или объекты. Автоматическое управление памяти подразумевает отсутствие такой необходимости и перенос ответственности за повторное использование памяти на дисциплину управления ею и соответствующую часть инфраструктуры программы – компилятор, менеджер памяти или специализированные библиотеки и функции. Например, метод подсчета ссылок относится к полуавтоматическому управлению памятью. Он работает по простой схеме: при создании или удалении новой ссылки на область памяти, происходит инкремент или декремент

ассоциированного с нею счетчика. Если счетчик равен 0, значит, данная область памяти больше не задействуется. Однако такой подход в чистом виде не позволяет работать с циклическими ссылками.

Алгоритмы сборки мусора относятся к методам автоматического управления памятью. Их идея состоит в просмотре используемых в программе структур данных, их классификации, на используемые и более не нужные. Последние считаются мусорными и подлежат освобождению для повторного использования, занимаемого ими пространства памяти. Рассмотрим основные варианты алгоритмов сборки мусора [1].

Идея трассирующего сборщика мусора заключается в обходе всей кучи по ссылкам, начиная от корней (в общем случае таковыми считаются регистры, стек и глобальные переменные), помечая посещенные объекты. Те объекты, которые не были помечены во время обхода, удаляются.

Для уплотняющего сборщика мусора к решаемым им задачам добавляется борьба с фрагментацией пространства кучи. Достигается это следующим образом: по мере достижения порога, запускающего сборку, начинается обход ссылок, в ходе которого достижимые объекты перемещаются к началу кучи, а ранее указывавшие на них ссылки, соответствующим образом модифицируются. В результате переживающие ряд сборок данные оказываются вблизи границы кучи и не мигрируют при последующих сборках.

Копирующий сборщик мусора в отличие от уплотняющего использует два региона кучи, обозначим их как первый и второй. Новые объекты аллоцируются в первом. Когда место в нем исчерпано, все доступные из корней объекты перемещаются во второй регион. В результате аналогично уплотняющему сборщику куча уплотняется, но ссылки модифицируются уже после этого. В конце алгоритма назначение регионов меняется: теперь аллокации производятся во втором, а первый – зарезервирован.

Сборка мусора с поколениями опирается на предположение о полярности времени жизни объектов, заключающейся в следующем: для новых объектов оно невелико, а объекты, уже пережившие несколько итераций сборки мусора, переживут их и далее. Вновь аллоцированные объекты относятся к первому поколению. Если такой объект переживает первую сборку мусора, он переквалифицируется во второе поколение, на следующей итерации – в третье, и так далее. Чем старше поколение, тем реже оно подвергается сборке мусора.

На практике рассмотренные выше алгоритмы применяются в симбиозе, модифицируются и оптимизируются. Рассмотрим некоторые реализации сборщиков мусора.

В .Net Framework используется Generational Mark&Sweep алгоритм. Все объекты разделяются на три поколения. Поколение 0 – короткоживущие объекты, поколение 1 – те, которые пережили сборку мусора, поколение 2 – долгоживущие. Сборщик мусора начинает работать, когда какое-либо поколение полностью заполнится. В зависимости от размера объектов в каждом поколении определяется, какие поколения затронет данная сборка мусора. Сборки мусора в более старшем поколении затрагивают и более молодые поколения. При этом в старшем поколении происходит уплотнение, а копирование происходит при перемещении объектов из нового в более старое [2].

Алгоритм Mark&Sweep состоит из двух фаз:

1. фаза сканирования: определяется, какие вещи в куче достижимы. Сюда входят указатели в стеках, регистры, глобальные переменные, и далее, указатели в куче;
2. фаза маркировки: проход по графу указателей. Помечаются объекты, как достижимые по ходу исполнения программы.

В .Net существуют два сборщика мусора, каждый из которых руководствуется определенными принципами. В обоих объекты, в зависимости от размера, попадают в SOH (Small Object Heap) или LOH (Large Object Heap). LOH не уплотняется.

Server GC (Global catalog) использует разделение кучи на сегменты, числом соответствующие количеству логических процессоров. Для обработки каждого сегмента выделяется по одному потоку.

Workstation mode. GC предназначен для работы на клиентских машинах. Он следит за нагрузкой CPU и старается работать с минимальными задержками. Хорошо подходит для работы в приложениях с UI.

Оба сборщика мусора доступны в двух режимах – параллельном и синхронном. При параллельном режиме GC запускается в отдельном потоке (с приоритетом normal) для второго поколения. При этом происходит полная остановка приложения (Stop The World, STW). Также если приложению доступно много свободной памяти, то SOH не уплотняется.

Во многих сборщиках мусора, в том числе и в сборщике мусора языка Go, используется абстракция «Триколор». Все объекты условно делятся на три типа: черные – достижимые, белые – еще не просмотренные или недостижимые, серые – достижимые, но ссылки которых еще не просмотрены.

Алгоритм сборщика мусора Go устроен так, чтобы позволить сборщику мусора работать без долгих STW. Для этого предусмотрен специальный барьер записи, чтобы новые объекты попадали в серую область.

Представим, что авторы изначально находятся в фазе Sweep и случилось некое событие, после которого было решено, что пора запускаться. После этого происходит переход в фазу Sweep Termination, происходит STW, и нужно завершить удаление всех остатков мусора, связанных с прежним sweep'ом. После можно перейти в фазу Mark и запустить выполнение алгоритма заново. Обходим корневые объекты и подготавливаемся к сканированию. Все, что нужно отсканировать, кладется в working pool. При этом нужна короткая пауза приложения, чтобы взять корни (они изначально помечаются как серые). Берем задачи из working pool'а, обходим серую зону, новые объекты добавляем в серую область. Дальше наступает момент, когда глобальная очередь пуста, но все не отсканировано до конца. Тогда останавливаем работу всех worker'ов GC, выталкиваем все кэши очередей в главную очередь и продолжаем обработку. На протяжении всех предыдущих фаз был поднят барьер для записи. После завершения пометки вновь запускаем приложение и начинаем sweep в фоне.

В Python (CPython) (рис. 1) используется reference counter и generational mark&sweep. Все объекты разделены на три поколения. Новые объекты относятся к самому молодому поколению – первому. Если они не отсеиваются в результате сборки мусора, их классификация меняется – теперь они относятся к следующему поколению. Обычно новые объекты являются короткоживущими, поэтому среди них сборка мусора проводится чаще, чем среди тех, которые уже пережили череду сборок мусора. С каждым поколением ассоциирован счетчик, хранящий разницу между количеством аллокаций и деаллокаций. При достижении определенного порога начинается сборка мусора.

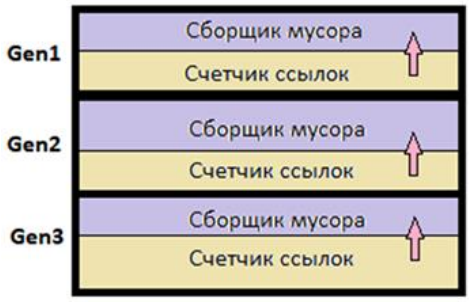


Рис. 1. Устройство сборщика мусора в Python

В JVM Hotspot реализовано четыре режима сборки мусора: Serial, Parallel, Concurrent Mark&Sweep и Garbage First (G1). Все они основываются на описанном ранее Generational Mark&Sweep алгоритме [3].

Serial подразумевает использование четырех регионов памяти – Eden, Survivor 0, Survivor 1 и Tenured. При сборке происходит STW, а сам алгоритм анализ памяти выполняется последовательно.

Parallel очень похож на Serial, но и малая, и полная сборки многопоточны. В малой сборке многопоточность используется для переноса объектов в старшее поколение, в полной – при уплотнении. При этом у каждого потока свой участок памяти в регионе Old Gen. Данный сборщик мусора работает с полной остановкой всего приложения. Также он может автоматически подстраивать параметры под тип поведения приложения.

CMS (Content Management System) GC имеет аналогичную организацию памяти. В нем возможны три режима сборки: в младшем поколении, в старшем и полный. Для сборки в старшем поколении к корням добавляются все объекты младшего поколения. В регионе Tenured сборщик мусора CMS работает в фоновом режиме постоянно. Алгоритм работы примерно следующий. Сначала происходит STW, и вычисляются корни. Далее приложение вновь запускается, и фаза пометки выполняется в параллельном потоке. После ее завершения работа приложения приостанавливается еще раз и выполняется просмотр объектов, инстанцированных за время пометки. При этом появляется «плавающий мусор» – уже не востребованные объекты, не выявленные за текущую итерацию сборки, они будут выявлены при следующей итерации. После завершения пометки приложение возобновляется, а сборщик приступает к фазе Sweep в фоновых потоках.

В Garbage-First организация памяти другая. Вся куча разбивается на много регионов одинакового размера. Каждый регион может быть Eden, Survivor, Tenured или Humongous (громадные регионы, которые не участвуют в сборке мусора и не перемещаются). Разделение на Eden, Survivor и Tenured условное. Каждый регион может выступать в роли любого из перечисленных. При этом малые сборки мусора выполняются не на всей куче, а только на части регионов, которые определяются как наиболее замусоренные. Перенос объектов в старшее поколение выполняется многопоточно с STW.

В Ruby Incremental Generational сборщик мусора. На фазе Sweep сборщик мусора не освобождает только количество памяти, необходимое для продолжения выполнения программы. При создании следующего объекта опять запускается сборщик мусора и освобождает минимально требуемый объем памяти. Такой алгоритм разделяет сборку мусора на несколько небольших процессов, которые чередуются с основным процессом приложения, что помогает избежать долгих пауз приложения.

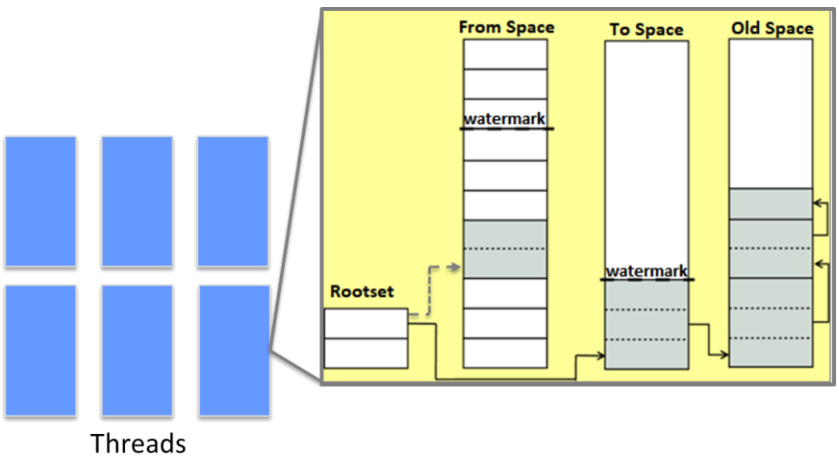


Рис. 2. Устройство сборщика мусора в Erlang

В Erlang Copying semi-space сборщик мусора (рис. 2). Причем в каждом легковесном потоке свой сборщик мусора. Это позволяет избежать полной остановки приложения. Происходят только паузы ниток. Вся куча делится на Yong и Old Heap. From Space и To Space размещаются в Young Heap’е. Сборка мусора начинается с прохода по Rootset’у и копирования всех объектов, на которые есть указатели из так называемого From Space в To Space пока все ссылки из rootset’а и уже скопированных в To Space объектов не станут указывать в To Space. Метка watermark – место, где закончилась предыдущая сборка мусора. В случае если объект находится под меткой watermark, он копируется не в To Space, а в Old Heap и ссылка на него

игнорируется при следующих сборках мусора. В случае полной сборки молодое и старое поколение имеют общий To Space. Она происходит, когда размер кучи под меткой больше размера оставшегося свободного места.

Для сравнения выбранных сборщиков мусора был проведен бенчмаркинг. Были написаны тесты, аллоцирующие 20 ГБ памяти небольшими сообщениями размером 1 КБ. Анализ полученных результатов дал следующие графики (рис. 3 и 4).

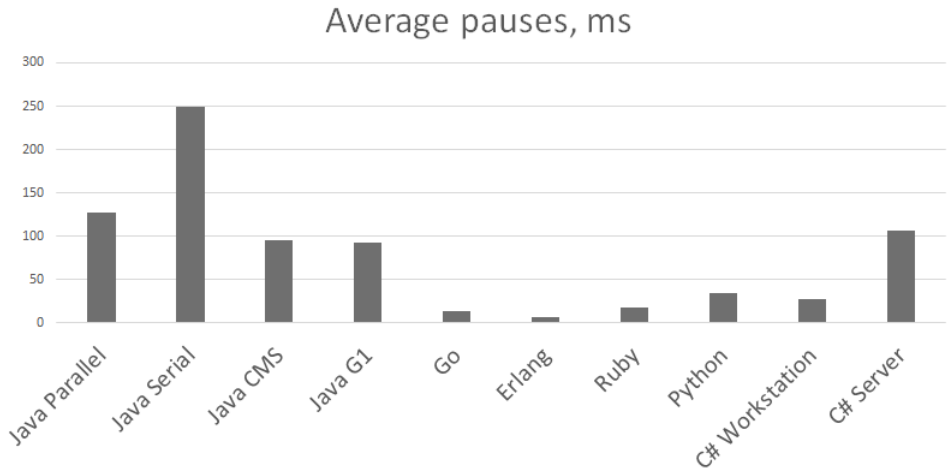


Рис. 3. Средняя продолжительность паузы приложения при сборке мусора

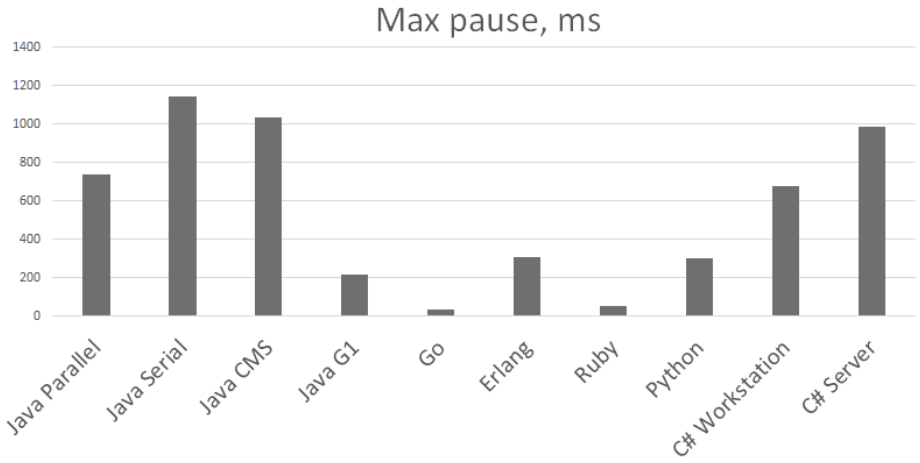


Рис. 4. Максимальная продолжительность паузы приложения при сборке мусора

Из полученных результатов можно составить рекомендации по использованию данных сборщиков мусора. Если не требуется большой размер кучи, приложение не чувствительно к коротким остановкам, и при этом доступно только одно ядро процессора, подойдет сборщик мусора Java Serial, если доступно несколько ядер и нужна автоматическая подстройка параметров сборщика мусора, Java Parallel. Если вашему приложению нужно обрабатывать большие объемы данных с хорошей пропускной способностью, подойдут Java CMS, C# Server и Ruby. В случае, когда не требуется короткоживущих объектов большого размера и не важна пропускная способность, следует выбрать Java G1. C# Workstation хорошо оптимизирован для работы на клиентских машинах, и приложений с UI. Если не требуется много долгоживущих объектов и не важна пропускная способность, можно выбрать Python. Если нужно сильно распараллеленное отказоустойчивое приложение, отлично подойдут сборщики мусора Erlang и Go [4].

Литература

1. Jones R., Hosking A., Moss E. The garbage collection handbook: the art of automatic memory management. – New York: Chapman and Hall/CRC, 2011. – 516 p.

2. Farrell C., Harrison N. Under the Hood of NET Memory Management. – Simple Talk Pub., 2011. – 238 p.
3. Memory Management in the Java HotSpot™ Virtual Machine [Электронный ресурс]. – Режим доступа: https://www.coursehero.com/file/22883726/memorymanagement-whitepaper-1-150020/ (дата обращения: 04.01.2019).
4. Ford R. Concurrent algorithms for real-time memory management // IEEE Software. – 1988. – V. 5. – № 5. – P. 10–23.

Кузнецова Наталья Валерьевна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4262
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: nvkuznetsova@list.ru

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.43

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОЛОГИЙ РАЗРАБОТКИ НА ПЛАТФОРМЕ PHP: АСИНХРОННОСТЬ В PHP

Кузнецова Н.В.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

В работе рассмотрена асинхронная методология программирования и ее отличия от синхронной. Описаны такие асинхронные технологии платформы PHP как ReactPHP, Amphp, Swoole. Дана сравнительная характеристика данных технологий и сделаны выводы о целесообразности их применения, а также сформулированы рекомендации, когда использование рассмотренных технологий может быть предпочтительнее. Дана краткая характеристика перспектив развития платформы PHP.
Ключевые слова: асинхронность, PHP, ReactPHP, Amphp, Swoole, цикл событий (event loop), promise, потоки.

Асинхронная концепция применяется во многих веб-платформах: JavaScript, C#, Python. Чаще всего с идеей асинхронности связывают именно язык JavaScript и платформу Node.js, которая позволяет писать эффективные в плане производительности и масштабируемые приложения. Платформа Node.js хорошо подходит для создания различных видов приложений: веб-приложения, приложения командной строки и демоны, настольные или десктопные приложения, а также приложения реального времени. Однако PHP, благодаря своей простоте и большому функционалу, долгое время оставался лидером в сфере back-end разработки. Данная технология в той или иной степени задействована более чем на 80% сайтов, которые до сих пор нуждаются в обслуживании и развиваются. Таким образом, может возникнуть необходимость внедрения асинхронного функционала в существующее PHP-приложение, что удобнее всего будет сделать именно на PHP, в чем могут помочь асинхронные PHP-библиотеки.

Изначально язык PHP основан на синхронной концепции, которая блокирует выполнение кода, пока не будет получен ответ на отправленный запрос. Избавиться от блокирующих операций позволяет асинхронность. Ключевая идея асинхронного программирования – возможность запускать вызовы методов и продолжать работу приложения, не дожидаясь окончания вызовов. Асинхронный ввод/вывод – это концепция обработки данных ввода/вывода, которая позволяет коду выполнять другие задачи, пока совершается обработка запроса. Асинхронные операции позволяют увеличить

производительность процессора за счет использования времени ожидания ответа на выполнение других задач, т.е. в асинхронном программировании задачи могут выполняться некоторыми дискретными порциями, освобождая вычислительные ресурсы процессора, когда они ожидают ответ на запрос. Асинхронный поток выполнения операций представлен на рисунке.

Рисунок. Асинхронный поток операций

Традиционный процесс запрос-ответ к PHP-приложению происходит по следующему сценарию: клиент отправляет запрос серверу, сервер запускает нужный PHP-скрипт, который возвращает данные серверу, а сервер, в свою очередь, отправляет их клиенту. Таким образом, получается, что для каждого запроса запускается отдельный процесс, который уничтожается, закончив работу. Этот подход сильно отличается от подхода Node, где приложение запущено в одном процессе и ждет, когда произойдет событие. В традиционной модели PHP роль слушателя событий перенесена непосредственно на веб-сервер, который при получении запроса запрашивает требуемый PHP-скрипт. Несложно заметить, что в данной модели тратятся ресурсы веб-сервера и временные ресурсы на запуск скрипта, получение ответа и завершение процесса. Асинхронная концепция также решает данную проблему. Язык PHP пока не располагает инструментами на уровне языка для написания асинхронных приложений, однако, предоставляет большой набор асинхронных библиотек и инструментов: ReactPHP, Amphp, Swoole, Kraken и др.

ReactPHP – это библиотека языка PHP, представляющая собой набор компонентов для событийного программирования на PHP [1]. К ключевым компонентам библиотеки относятся цикл событий (event loop), promise и потоки.

Цикл событий – это программная конструкция, которая ожидает возникновения и производит рассылку событий или сообщений в исполняемой программе. Event loop в ReactPHP не скрыт, поэтому необходимо явно его объявлять [1]. Event loop выполняется до тех пор пока циклу есть, что обрабатывать. Когда все инициализированные обработчики событий выполнены, event loop останавливается и управление кодом возвращается в синхронный режим.

Promise в ReactPHP основаны на двух концепциях: deferred, promise [1]. Deferred представляет собой единицу работы или вычислений, которая, возможно, еще не завершена или еще не начата. Данная работа или вычисление выполняется асинхронно и завершается в некоторый момент времени в будущем. Если deferred представляет собой само вычисление, то promise – это результат этого вычисления. В ReactPHP promise обычно реализуется именно через deferred. На promise в ReactPHP основана работа таких компонентов как DNS и Filesystem [1].

Потоки позволяют упростить использование цикла событий. В ReactPHP потоки похожи на реализацию потоков PHP на уровне языка, но адаптированы для асинхронного неблокирующего ввода/вывода [1]. В ReactPHP доступны реализации потоков для чтения, записи и дуплексные. Также ReactPHP поддерживает возможности пайпинга. Пайпинг – это механизм, средствами которого выходные данные одного потока передаются в качестве входных другому.

Amphp – это неблокирующий PHP-фреймворк [2]. Как и ReactPHP предоставляет ключевые компоненты для написания асинхронного кода: event loop, потоки и promise. Сочетая promise с генераторами, фреймворк формирует корутины, позволяющие писать асинхронный код синхронным способом, т.е. без необходимости использовать коллбэк-функции [2]. Корутины являются прерываемыми функциями, похожими на генераторы. С помощью генераторов обычно реализуются простые итераторы, и создаются значения с

помощью ключевого слова yield. Amp использует yield как точку прерывания. Когда в coroutine-функции возникает значение, выполнение функции прерывается, и выполняются другие задачи (обработчики ввода/вывода, другие функции-корутины) [2]. Функцию-корутину от генератора отличает возможность передавать параметры (с помощью метода send()) из внешнего кода каждый раз, когда функция выбрасывает очередное значение.

Swoole – высокопроизводительный фреймворк, написан на языке C как расширение PHP [3]. Используя Swoole можно писать высокопроизводительные и масштабируемые TCP, UDP, HTTP, WebSocket сервисы. В отличие от Nginx, Tornado, и Node.js, Swoole имеет встроенные асинхронные многопоточные модули ввода/вывода. Как и Amphp, фреймворк основан на корутинах. Swoole создает одну coroutine-функцию для каждого запроса и переключается между ними, основываясь на статусе ввода/вывода [3].

В таблице приведено сравнение рассмотренных асинхронных PHP-библиотек по некоторым критериям.

Таблица. Сравнение асинхронных библиотек PHP

	ReactPHP	Amphp	Swoole
Версия PHP	PHP 5.4 и выше	PHP 7.0	PHP 5.4 и выше
Базы данных	MySQL, Redis, MongoDB	MySQL, Postgres, Redis	MySQL
Документация	Отличная, с множеством примеров	Менее подробная, чем ReactPHP	Недостаточно полная, мало примеров
Сложность установки	Легкая	Легкая	Сложная
Язык исходного кода	PHP	PHP	C, C++

Язык PHP предоставляет достаточно большой набор инструментов для написания асинхронных приложений с неблокирующим вводом/выводом. Наиболее проста в изучении – библиотека ReactPHP. Данная технология предоставляет функционал, аналогичный функционалу языка JavaScript и платформы Node.js. Однако, несмотря на эффективность платформы Node.js, есть ситуации, когда платформе Node лучше предпочесть другую технологию. В PHP-приложении это может быть необходимость оптимизировать затраты веб-сервера и временные ресурсы. В случае необходимости опытный PHP-разработчик скорее освоит новую технологию знакомого и более привычного языка, чем перейдет на новую платформу.

Таким образом, можно сформулировать следующие рекомендации. Использовать асинхронные PHP-технологии можно в случаях:

1. необходимости внедрения в PHP-приложение асинхронного функционала, чтобы не добавлять в приложение другую платформу, основанную на менее знакомой PHP-разработчику технологии, что позволит упростить поддержку приложения;
2. необходимости написания PHP-приложения с неблокирующим вводом/выводом;
3. если PHP-разработчик желает повысить свою квалификацию. Для новичков в асинхронных библиотеках PHP лучше всего подойдет именно ReactPHP с лучшей документацией и более развитым сообществом;
4. в исследовательских целях для поиска альтернатив платформе Node.js.

Рассматривая перспективы асинхронности в PHP, следует отметить заинтересованность разработчиков во внедрении нового функционала в язык. Один из разработчиков языка PHP Зеев Сураски в июне 2018 года опубликовал письмо, в котором поделился планами команды в отношении языка и перспективах. Главная новость данного письма – это готовность команды разработчиков завершать линейку PHP7 и переходить к разработке новой версии языка PHP8 [4]. В планах на новую версию внедрение компиляции на лету (JIT), которая позволит PHP улучшить свою производительность в рамках разработки, выходящей за пределы веба, а также асинхронность [4]. В письме говорится, что необходимо добавить больше поддержки

операций асинхронного ввода/вывода непосредственно в сам язык, чтобы расширить круг задач, где в настоящее время имеет смысл применять такие инструменты как ReactPHP, Amphp или Swoole [4]. Таким образом, платформа PHP развивается и готова выйти на новый этап своего жизненного цикла. Внедрение асинхронности в PHP позволит платформе дольше оставаться востребованной на рынке веб-разработки, а также внедрять новый функционал в уже существующие PHP-приложения.

Литература

1. Event-driven, non-blocking I/O with PHP [Электронный ресурс]. – Режим доступа: https://reactphp.org (дата обращения: 04.01.2019).
2. Amp – Asynchronous concurrency made simple [Электронный ресурс]. – Режим доступа: https://amphp.org/amp (дата обращения: 04.01.2019).
3. Swoole Documents [Электронный ресурс]. – Режим доступа: https://www.swoole.co.uk/docs (дата обращения: 04.01.2019).
4. PHP 2^3 [Электронный ресурс]. – Режим доступа: https://externals.io/message/102415 (дата обращения: 04.01.2019).

Кузьмин Кирилл Сергеевич
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4222
Направление подготовки: 09.04.04 – Программная инженерия систем
реального времени
e-mail: wee3dood@gmail.com

Штенников Дмитрий Геннадьевич
Год рождения: 1975
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: dshtennikov@gmail.com

УДК 004

МЕТОД ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ПЕРЕДАЧИ ДАННЫХ ПРИ ОБНОВЛЕНИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОБИЛЬНЫХ УСТРОЙСТВ

Кузьмин К.С.
Научный руководитель – к.т.н., доцент Штенников Д.Г.

Работа содержит описание метода обеспечения безопасности передачи данных при обновлении программного обеспечения мобильных устройств, а также описание его программной реализации.
Ключевые слова: целостность ПО, безопасность передачи данных, мобильные устройства.

Введение. Одной из проблем в современном мире является вопрос целостности программного обеспечения (ПО) различных устройств. Одним из возможных решений для журналирования и контроля целостности ПО может стать технология блокчейн. Однако использование данной технологии на мобильных устройствах осложнено из-за их низкой производительности, а также из-за ограниченного объема памяти [1–3].

Для решения данной проблемы предлагается прибегнуть к комбинации технологий: использовать приватный блокчейн на производительных серверных машинах для обеспечения контроля целостности и журналируемости системы, а также контроль целостности ПО при обновлении ПО мобильных устройств при помощи хеширования и обеспечениие безопасности передачи таких данных при помощи шифрования.

В данной работе рассмотрен процесс взаимодействия клиента, представленного в виде мобильного устройства с низкой производительностью, и серверной машины, в качестве которой будет выступать устройство, с достаточной для поддержания приватного блокчейна производительностью.

Архитектура системы. Сама система передачи состоит из модулей сборки информации об устройстве и хеша ПО, шифрования полученных данных для обеспечения безопасности их передачи, а также передача самих данных.

Благодаря сочетанию технологий хеширования и шифрования обеспечивается контроль целостности и безопасность передачи данных об устройствах, а система верификации позволяет обеспечить журналируемость данных об устройствах и легко отследить изменения при сбоях или вредоносных изменениях ПО извне.

Система состоит из нескольких частей:

– блок хеширования;
– блок шифрования;
– блок транспорта.

Блок хеширования – использование классического метода хеширования для создания «слепка» бинарных файлов системы или программных файлов, позволяющих проверить целостность ПО.

Блок шифрования – для обеспечения безопасности данных на транспорте используется шифрование. Хеш-свертка вместе с дополнительными данными «оборачиваются» в шифрование, позволяя обезопасить хеш от изменения на транспорте.

Блок транспорта – данные, полученные в результате работы предыдущих двух блоков, необходимо передать. Для этого создается клиент, подключающийся по сети с сервером и передающий ему данные.

Архитектура системы представлена на рис. 1.

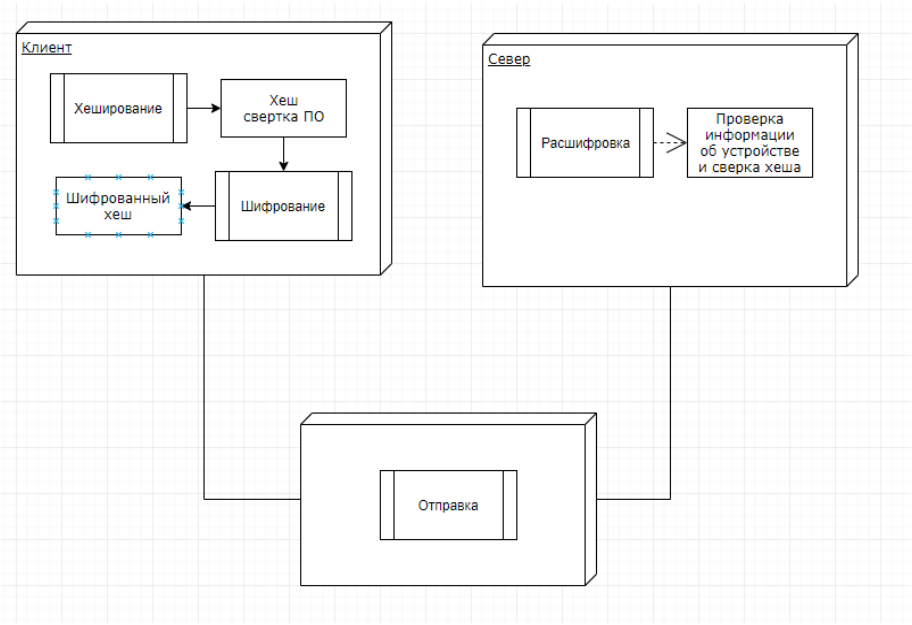


Рис. 1. Архитектура системы

Система должна представлять собой три блока, каждый из которых содержит свою компонентную реализацию. На рис. 1 представлены блоки системы, и то, по какому пути они взаимодействуют между собой.

Система работает по следующему принципу:

1. происходит запрос к клиентскому устройству;
2. устройство возвращает идентификационные данные об устройстве;
3. устройство собирает хеш заранее заданных файлов/информации и дополнительные параметры устройства;
4. происходит шифрование данных;
5. данные передаются серверу;
6. сервер расшифровывает данные.

Диаграмма классов выглядит следующим образом (рис. 2).

Рассмотрим диаграмму классов подробнее.

Существует два основных класса – клиент и сервер. Они осуществляют взаимодействие друг другом путем сетевых подключений и передачи данных и запросов друг другу.

Клиент (Client) – класс, позволяющий клиентской машине осуществлять взаимодействие с серверным устройством и обеспечивающий передачу данных.

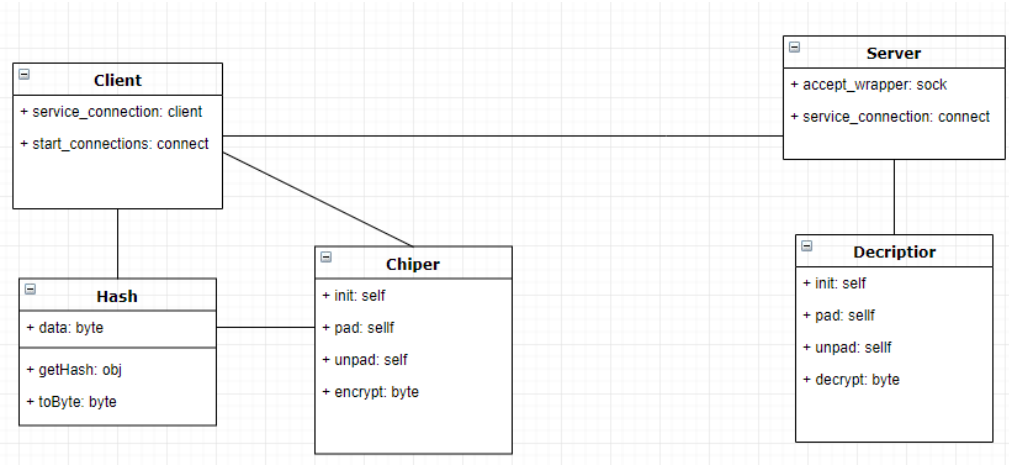


Рис. 2. Диаграмма классов проектирования

На стороне клиента также работают: класс сбора хеш-сумм ПО (Hash) и класс шифрования данных (Chiper).

Сервер (Server) работает на серверной машине, содержащей систему журналирования и верификации ПО.

Класс дешифрования данных (Decriptor) аналогичен клиентскому классу шифрования и служит для дешифровки полученных данных.

Сервер после получения и обработки данных передает их из системы верификации.

Итоговый процесс выглядит следующим образом (рис. 3).

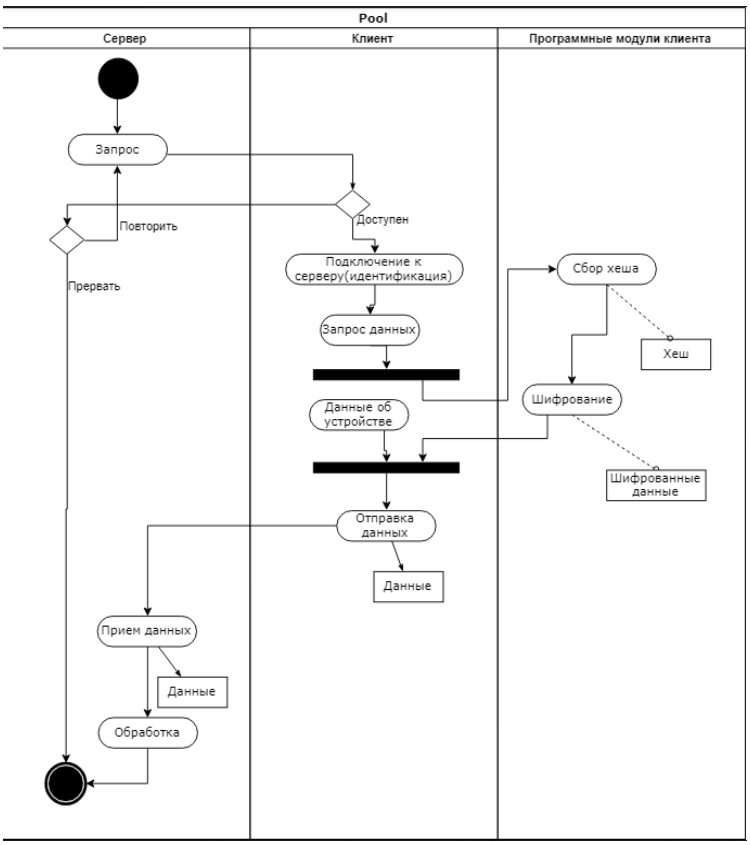


Рис. 3. Диаграмма деятельности

Выводы. Таким образом, спроектирована и разработана система, позволяющая обеспечить безопасность данных об устройстве и его ПО при передаче данных и обновлении ПО. В дальнейшем планируется произвести тестирование и оценку работы всей системы в целом с различными параметрами программной и аппаратной части.

Литература

1. Домарев В.В. Безопасность информационных технологий. Методология создания систем защиты. – Киев: ООО «ТИД «ДС», 2002. – 688 с.
2. Блокчейн и IoT: перспективы взаимодействия и проблемы на пути развития [Электронный ресурс]. – Режим доступа: https://forklog.com/blokchejn-i-iot-perspektivy-vzaimodejstviya-i-problemy-na-puti-razvitiya/ (дата обращения: 04.01.2019).
3. Проблемы и перспективы интернета вещей [Электронный ресурс]. – Режим доступа: https://rb.ru/opinion/russian-iot/ (дата обращения: 04.01.2019).

Лавров Алексей Валерьевич
Год рождения: 1986
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: lavrov@corp.ifmo.ru

Сизиков Валерий Сергеевич
Год рождения: 1939
Университет ИТМО, факультет программной инженерии и компьютерной техники, д.т.н., профессор
e-mail: sizikov2000@mail.ru

УДК 004.932, 519.642.3

ОЦЕНКА ПОГРЕШНОСТИ ВОССТАНОВЛЕНИЯ ЧАСТИЧНО ПЕРЕКРЫВАЮЩИХСЯ СПЕКТРАЛЬНЫХ ЛИНИЙ

Лавров А.В.
Научный руководитель – д.т.н., профессор Сизиков В.С.

Работа выполнена в рамках темы НИР № 619296 «Разработка методов создания и внедрения киберфизических систем».

Рассмотрена задача разделения (сепарации) перекрывающихся линий в непрерывном спектре. Спектральные линии в данной работе моделируются лоренцианами. Количество линий и приближенные значения их параметров (центральная длина волны, амплитуда и полуширина) определяются с помощью метода производных, при этом зашумленный суммарный спектр предварительно сглаживается сплайном для обеспечения устойчивости его численного дифференцирования. Уточнение параметров линий производится путем минимизации функционала невязки между экспериментальным и рассчитанным спектрами. С помощью компьютерного моделирования в пакете MATLAB проводится анализ зависимости погрешности восстановления параметров линий от различных условий: уровня шума, настроек алгоритмов сглаживания, соотношений параметров перекрывающихся линий.
Ключевые слова: восстановление спектра, численное дифференцирование, MATLAB.

Введение. Анализ параметров отдельных спектральных линий используется для изучения ряда физических явлений, таких как эффект Зеемана, эффект Штарка, межмолекулярное взаимодействие, соударение частиц, и др. При этом близко расположенные компоненты спектра могут частично перекрываться, не позволяя явным образом определить параметры линий, а порой даже не позволяя однозначно определить, какое количество компонент имеется в суммарном спектре. Задача сепарации перекрывающихся спектральных линий рассматривается в различных работах, где предлагаются: метод дифференцирования спектра [1], метод Фурье-самодеконволюции [2], метод свертки измеренного спектра с производными гауссовых функций [3], метод вычисления моментов высших порядков измеренного спектра [4], и др. В работе [5] на основе анализа предшествующих подходов предлагается комплексный алгоритм восстановления искаженного непрерывного спектра и его компонент, вводится методика оценки качества восстановления по функционалу невязки

между измеренным и восстановленным спектром, а также по взвешенному среднеквадратическому отклонению между восстановленными параметрами линий спектра и истинными (известными в случае решения задачи на модельных примерах).

Цель данного исследования: оценка погрешности δ**p** восстановления спектральных линий по описанному в работе [5] алгоритму и анализ зависимости погрешности от различных параметров задачи. Для этого используется компьютерное моделирование: решается прямая задача формирования зашумленного суммарного спектра $\tilde{Z}$ из N линий z_j, $j = \overline{1, N}$ (т.е. моделируется экспериментальный спектр), а затем решается обратная задача (восстановление по экспериментальному спектру параметров его линий). При этом варьируются как условия прямой задачи (уровень шума δz, величина шага дискретизации h, значения параметров каждой j-ой линии спектра: амплитуда A_j, полуширина τ_j, центральная длина волны $\bar{\lambda}_j$), так и условия обратной задачи (параметры сглаживания экспериментального спектра для устранения шума, параметры используемых алгоритмов минимизации функционала невязки), чтобы проследить, как эти условия влияют на погрешность восстановления.

Описание прямой задачи. Спектральные линии z_j в данной работе моделируются лоренцианами (в работах [5, 6] также рассмотрен случай моделирования линий гауссианами). Лоренцева функция задается тремя параметрами – амплитуда A_j, полуширина τ_j, центральная длина волны $\bar{\lambda}_j$:

$$z_j(\lambda) = A_j \frac{\tau_j^2}{(\lambda - \bar{\lambda}_j)^2 + \tau_j^2}, \quad j = \overline{1, N}. \tag{1}$$

Экспериментальный спектр $\tilde{Z}(\lambda)$ моделируется как сумма N линий-компонент $z_j(\lambda)$ и случайного гауссова шума δz:

$$\tilde{Z}(\lambda) = \sum_{j=1}^{N} z_j(\lambda) + \delta z, \quad a \le \lambda \le b, \tag{2}$$

где a, b – границы по λ экспериментального спектра.

Моделируемый экспериментальный спектр численно представляется как массив отсчетов по дискретной сетке λ от a до b с шагом дискретизации h.

Описание обратной задачи. Для определения количества линий-компонент используется численное дифференцирование суммарного экспериментального спектра $\tilde{Z}(\lambda)$. Центральные длины волн $\bar{\lambda}_j$ линий-компонент обнаруживаются в тех точках, где производная третьего порядка от суммарного спектра пересекает нулевую ординату на подъеме. Два других параметра каждой линии A_j и τ_j вычисляются по формулам

$$A_j = 6\frac{[z_j''(\bar{\lambda}_j)]^2}{z_j^{IV}(\bar{\lambda}_j)}, \quad \tau_j = \sqrt{12\frac{-z_j''(\bar{\lambda}_j)}{z_j^{IV}(\bar{\lambda}_j)}}, \quad j = \overline{1, N}, \tag{3}$$

выведенным из решения системы линейных уравнений, полученных путем аналитического дифференцирования функции (1) в работе [6].

При этом экспериментальный спектр предварительно аппроксимируется кубическим сглаживающим сплайном, если он зашумлен. Для сглаживания используется m-функция csaps.m пакета MATLAB, степень сглаживания задается параметром $P \in [0,1]$. Заметим, что параметр P обратно пропорционален параметру регуляризации α [7, С. 223]: чем меньше P и чем больше α, тем больше сглаживание сплайна.

Уточнение параметров линий при решении обратной задачи. Параметры линий, вычисленные по формулам (3), оказываются, как правило, неточными, потому что на значения производных сильно влияют не только случайные шумы, но и профили соседних наложившихся линий. Поэтому для уточнения параметров рассчитывается суммарный спектр:

$$Z(\lambda) = \sum_{j=1}^{N} z_j(\lambda), \quad a \le \lambda \le b, \tag{4}$$

и производится минимизация функционала невязки F между экспериментальным спектром $\tilde{Z}$ и восстановленным суммарным спектром Z:

$$F = \sum_{i=1}^{n} \left(\tilde{Z}_i - Z_i\right)^2, \tag{5}$$

где $n=(b-a)/h+1$ – количество отсчетов по сетке λ от a до b с шагом h.

В данном исследовании рассмотрена минимизация с помощью симплекс-метода Нелдера–Мида, реализованного в пакете MATLAB в m-функции fminsearch.m. Работа этого метода регулируется параметрами TolFun (конечное допустимое отклонение по значению функции) и TolX (конечное допустимое отклонение по значению любого из параметров).

Оценка погрешности решения. Вычисленные значения параметров N линий объединяются в единый вектор $\mathbf{p}$ длиной $3N$, имеющий вид

$$\mathbf{p} = [A_1, \bar{\lambda}_1, \tau_1, \ldots, A_N, \bar{\lambda}_N, \tau_N]. \tag{6}$$

Погрешность решения $\delta\mathbf{p}$ рассчитывается по формуле

$$\delta\mathbf{p} = \left[\frac{1}{3N}\sum_{J=1}^{3N} w_J(\mathbf{p}_J - \bar{\mathbf{p}}_J)^2\right]^{1/2}, \tag{7}$$

где $\mathbf{p}_J$ – вычисленные значения параметров линий; $\bar{\mathbf{p}}_J$ – точные значения параметров (известные при обработке модельных спектров); w_J – весá, равные $w_J = 1/\bar{\mathbf{p}}_J^2$.

Степень наложения линий в данной работе характеризуется безразмерной величиной D, формула для которой в случае любого количества N линий предложена в [6]. Степень наложения для двух соседних линий (j-й и j+1-й) рассчитывается по формуле

$$D_{j,j+1} = \frac{\bar{\tau}_{j,j+1}}{\Delta_{j,j+1}}\left(1 + \frac{|A_j - A_{j+1}|}{A_j + A_{j+1}}\right), \qquad j = \overline{1, N-1}, \tag{8}$$

где $\bar{\tau}_{j,j+1} = (\tau_j + \tau_{j+1})/2$ – средняя полуширина соседних линий, а $\Delta_{j,j+1} = \bar{\lambda}_{j+1} - \bar{\lambda}_j$ – расстояние между центрами соседних линий.

Для всего суммарного спектра рассчитывается средняя степень наложения линий:

$$\bar{D} = \frac{1}{N-1}\sum_{j=1}^{N-1} D_{j,j+1}. \tag{9}$$

Численные иллюстрации. На рис. 1 показано несколько модельных примеров, на которых рассматривалась зависимость погрешности результата восстановления $\delta\mathbf{p}$ от уровня шума δz и параметра сглаживания P.

Первая модель (рис. 1, а) задана вектором $\bar{\mathbf{p}}$ (6) со значениями [5, 560, 5, 5, 590, 5], а вторая (рис. 1, б) – вектором [1, 570, 2, 2, 580, 5]. Поскольку во втором примере амплитуды линий меньше, чем в первом, то и отношение уровня сигнала к шуму тоже меньше.

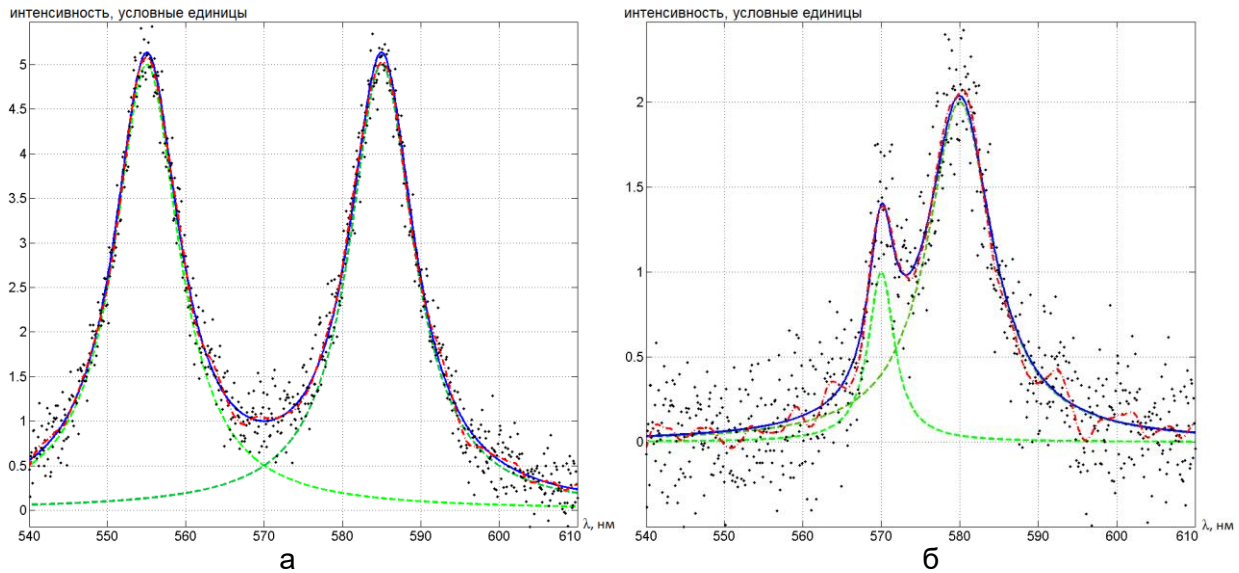


Рис. 1. Модельные спектры: две линии с равными A и τ и с малой степенью наложения ($\overline{D}$ =0,17) (а); две линии с неравными A и τ и с большей степенью наложения ($\overline{D}$ =0,47) (б). Зеленый пунктир – исходные точные профили линий; сплошная синяя линия – точный суммарный спектр; точки – зашумленный экспериментальный спектр (при δz=0,25); красный штрих-пунктир – сглаженный сплайном экспериментальный спектр (при δz=0,25 и P=0,5)

На рис. 2 показаны зависимости погрешности $\delta\mathbf{p}$ от уровня случайного гауссова шума δz для неуточненных значений параметров, вычисляемых по формулам (3), и для решений, полученных после минимизации функционала (5) методом Нелдера–Мида. Поскольку погрешность каждого отдельного измерения сильно меняется при различном случайном шуме одинакового уровня, то для усреднения было проведено по 300 измерений погрешности $\delta\mathbf{p}$ при каждом значении уровня шума δz. Величина $\delta\mathbf{p}$ на графике представляет собой среднее гармоническое от 300 измерений.

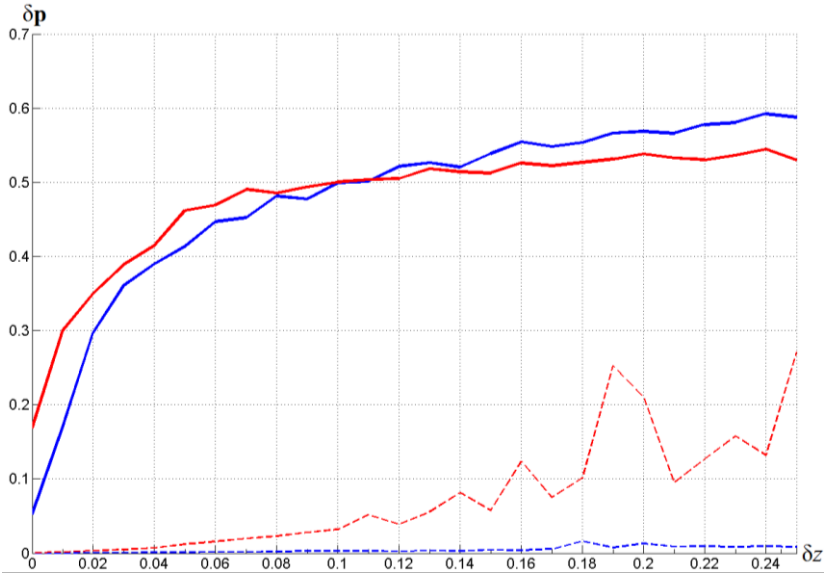
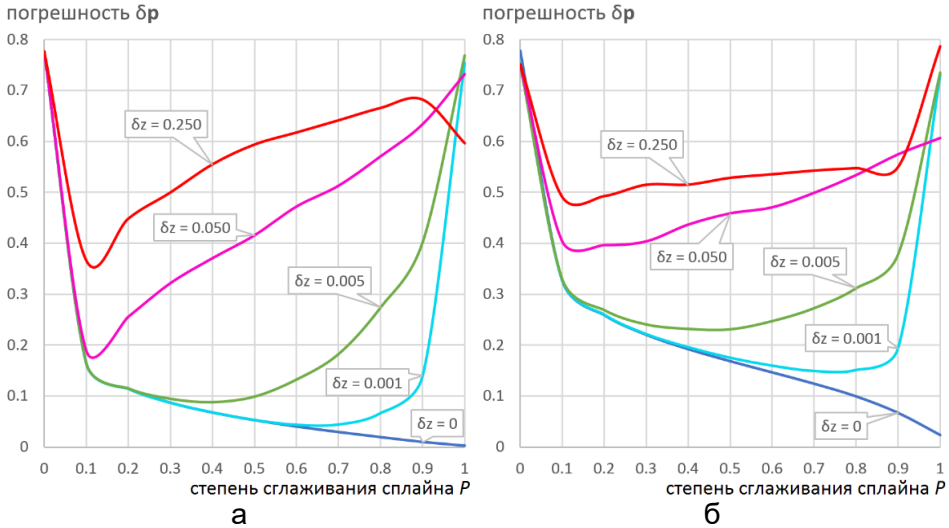


Рис. 2. Зависимость погрешности $\delta\mathbf{p}$ от уровня шума δz в случае модельного примера «а» (синяя кривая) и модельного примера «б» (красная кривая) при использовании сглаживающего сплайна с параметром P=0,5: сплошные кривые показывают погрешность неуточненного решения, а пунктир – уточненного методом Нелдера–Мида (с параметром tolX=0,001)

Как видно по графику на рис. 2, после уточнения параметров линий погрешность заметно снижается (особенно резкое снижение погрешности достигается при небольших значениях шума). Тем не менее, следует отметить, что при высоком уровне шума результат

«уточнения» параметров линий путем минимизации функционала может иметь иногда даже более высокую погрешность, чем исходное решение. Это связано с тем, что сильный уровень шума создает большое количество «ложных» пиков в сглаженном сплайном спектре (такие «ложные» пики хорошо видны на рис. 1, б), и алгоритм минимизации может сходиться к неверному решению, принимая ложный пик за истинный. Этим объясняются сильные флуктуации измерений δ**p**, особенно заметные на рис. 2 на красной пунктирной кривой при сильном уровне шума.

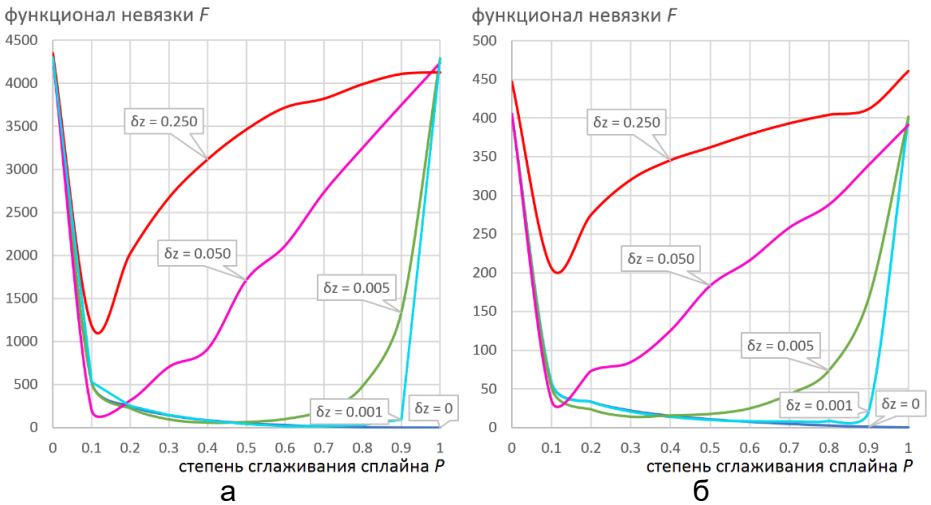
Зависимость погрешности δ**p** от степени сглаживания экспериментального спектра кубическим сплайном *P* показана на рис. 3. Опять же было произведено по 300 измерений погрешности при каждом значении *P* от 0 до 1 с шагом 0,1, и величина δ**p** на графике представляет собой среднее гармоническое от 300 измерений.



а б

Рис. 3. Зависимость погрешности неуточненного решения δ**p** от степени сглаживания сплайном *P* при разных уровнях шума δ*z*: для примера, показанного на рис. 1, а (а); для примера, показанного на рис. 1, б (б)

На рис. 4 продемонстрирована зависимость функционала невязки *F* от степени сглаживания *P*, посчитанная аналогичным образом (среднее гармоническое от 300 измерений). По рис. 3, 4 можно сделать вывод, что при наличии небольшого шума (δ*z*<0,005) значение параметра *P*=0,5 является близким к оптимальному. При более сильном шуме лучше использовать более сильное сглаживание (*P*<0,5).



а б

Рис. 4. Зависимость функционала невязки *F* между восстановленным неуточненным спектром *Z* и экспериментальным спектром $\tilde{Z}$ при разных уровнях шума δ*z*: для примера, показанного на рис. 1, а (а); для примера, показанного на рис. 1, б (б)

Заключение. В результате проведенного исследования было установлено, что зависимость погрешности $\delta\mathbf{p}$ неуточненного решения от уровня шума δz имеет экспоненциальный вид. Погрешность решения, уточненного методом Нелдера–Мида, в целом ниже, чем неуточненного, но при сильном наложении линий и большом уровне шума зависимость становится сильно неустойчивой.

Также было выявлено, что зависимость погрешности $\delta\mathbf{p}$ от степени сглаживания спектра сплайном имеет, как правило, один выраженный минимум, что позволяет подбирать оптимальное значение параметра сглаживания P для разного уровня шума.

Дальнейшим развитием исследования будет рассмотрение зависимостей на других моделях спектра (содержащих три и более спектральных линии), рассмотрение и сравнение различных методов минимизации функционала невязки, а также изучение влияния на погрешность других условий (величина шага дискретизации h, параметры используемых алгоритмов минимизации, и др.).

Литература

1. Giese A.T., French C.S. The analysis of overlapping spectral absorption bands by derivative spectrophotometry // Applied Spectroscopy. – 1955. – V. 9. – № 2. – P. 78–96.
2. Kauppinen J.K., Moffatt D.J., Mantsch H.H., Cameron D.G. Fourier self-deconvolution: a method for resolving intrinsically overlapped bands // Applied Spectroscopy. – 1981. – V. 35. – № 3. – P. 271–276.
3. Сирвидас С.И., Заруцкий И.В., Ларионов А.М., Манойлов В.В. Обнаружение, разделение и оценка параметров масс-спектрометрических пиков методом свертки экспериментальных данных с производными гауссовых функций // Научное приборостроение. – 1999. – Т. 9. – № 2. – С. 71–75.
4. Манойлов В.В., Заруцкий И.В. Алгоритмы оценки наличия двух и более пиков в выборке данных фрагмента масс-спектра для программного обеспечения обработки масс-спектров в условиях недостаточного разрешения // Научное приборостроение. – 2009. – Т. 19. – № 4. – С. 109–114.
5. Сизиков В.С., Лавров А.В. Современные устойчивые математические и программные методы восстановления искаженных спектров // Научно-технический вестник информационных технологий, механики и оптики. – 2018. – Т. 18. – № 6. – С. 911–931.
6. Сизиков В.С., Лавров А.В. Сравнение различных методов разделения непрерывных перекрывающихся спектральных линий // Оптика и спектроскопия. – 2018. – Т. 124. – № 6. – С. 723–731.
7. Сизиков В.С. Математические методы обработки результатов измерений. – СПб.: Политехника, 2001. – 240 с.

Лаврова Анна Владимировна
Год рождения: 1986
Университет ИТМО, центр юзабилити и смешанной реальности, инженер
e-mail: lavrova@corp.ifmo.ru

Спиридонова Анна Михайловна
Год рождения: 1982
Университет ИТМО, факультет программной инженерии и компьютерной техники, ст. преподаватель
e-mail: spiranna@list.ru

Лавров Алексей Валерьевич
Год рождения: 1986
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: lavrov@corp.ifmo.ru

Смолин Артем Александрович
Год рождения: 1977
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.философ.н.
e-mail: smolin@corp.ifmo.ru

УДК 004.5, 004.946, 069.295

ИНТЕРФЕЙС ПЕРЕМЕЩЕНИЯ МЕЖДУ МУЗЕЙНЫМИ ПРОСТРАНСТВАМИ В СИСТЕМАХ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Лаврова А.В., Спиридонова А.М., Лавров А.В.
Научный руководитель – к.философ.н. Смолин А.А.

Работа выполнена в рамках гранта РФФИ № 17-04-12034 «Мультимедийная информационная система «Архитектурно-художественный комплекс Феодоровский городок в Царском селе как пример Русского стиля».

В работе рассмотрены технологии и приемы реализации пользовательского интерфейса в виртуальном музее русского стиля. Основная экспозиция музея представлена в 4-х виртуально реконструированных помещениях Трапезной палаты Феодоровского городка в Царском селе. Технически каждое помещение реализовано в виде отдельной трехмерной сцены, поэтому рассматривалась задача переключения между сценами с учетом удобства и естественности восприятия пространства пользователем. В результате работы было создано приложение для просмотра в шлеме виртуальной

реальности, позволяющее пользователю в свободном порядке перемещаться между всеми помещениями виртуального музея.
Ключевые слова: виртуальный музей, реконструкция интерьера, WebGL.

Одним из актуальных направлений Digital Humanities является создание интерактивных виртуальных музеев [1]. В 2015 году на факультете программной инженерии и компьютерной техники Университета ИТМО началась работа по виртуальной реконструкции интерьеров и экстерьеров Феодоровского городка [2]. В дальнейшем было решено продолжить проект в виде создания виртуального музея русского стиля с экспонатами, представленными в смоделированных интерьерах Феодоровского городка. В основу музейного маршрута была положена реальная экскурсия Николая II по Трапезной палате в 1917 году. Возможные пути перемещения между помещениями представлены на рис. 1.

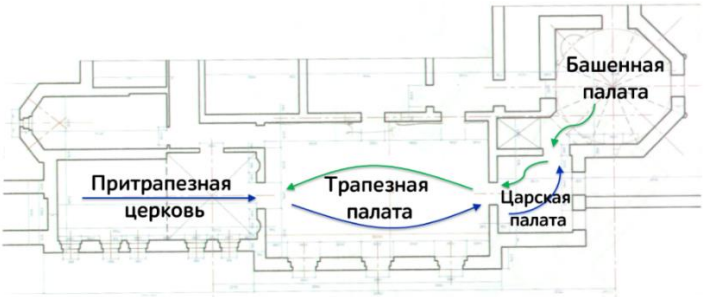


Рис. 1. Маршруты перемещения между залами виртуального музея русского стиля: зелеными стрелками показан прямой экскурсионный маршрут, а синими – обратный

В процессе создания виртуального музея были опробованы различные способы перемещения в пространстве для разных платформ: веб-браузера, гарнитур Homido, HTC Vive, Samsung Gear VR, Samsung Odyssey (рис. 2). В версии приложения для гарнитуры Homido было реализовано управление перемещением при помощи поворотов головы, так как другие средства взаимодействия в этой гарнитуре отсутствуют: направляя взгляд на интерактивные объекты (двери и окна), пользователь может приблизиться к ним [3]. В версии для браузера направление камеры меняется при помощи мыши. В гарнитуре Gear VR пользователь может перемещаться, используя кнопки контроллера, а HTC Vive и Samsung Odyssey дают возможность перемещаться не только с помощью контроллера, но и естественным образом.

а б в г

Рис. 2. Гарнитуры виртуальной реальности, для которых были разработаны различные версии приложения: Cardbox/Homido (а); Samsung Gear VR (б); HTC Vive (в); Samsung Odyssey (г)

Так как основной задачей работы являлась исторически достоверная реконструкция помещений, то интерьеры моделировались с высокой степенью детализации архитектурных элементов и высоким качеством текстур, что создает большую нагрузку на производительность компьютера при отображении моделей помещений. В связи с этим было решено загружать помещения музея по отдельности, добавляя в каждый момент времени на сцену только тот зал, в котором пользователь находится в данный момент. При переходе в соседний зал двери остаются закрытыми. Для того чтобы смена интерьера в этот момент

воспринималась пользователем более естественно, во время перемещения экран плавно затемняется. В версии приложения, использующей контроллер, переход через дверь осуществляется только по нажатию кнопки контроллера, чтобы исключить случайное попадание в соседний зал при приближении пользователя к двери.

Для удобства взаимодействия пользователя с интерактивными элементами приложения на сцене отображается указатель (рис. 3), который может менять положение и направление в соответствии с перемещением контроллера. Указатель подсвечивается при наведении на интерактивные элементы (двери и экспонаты музея).

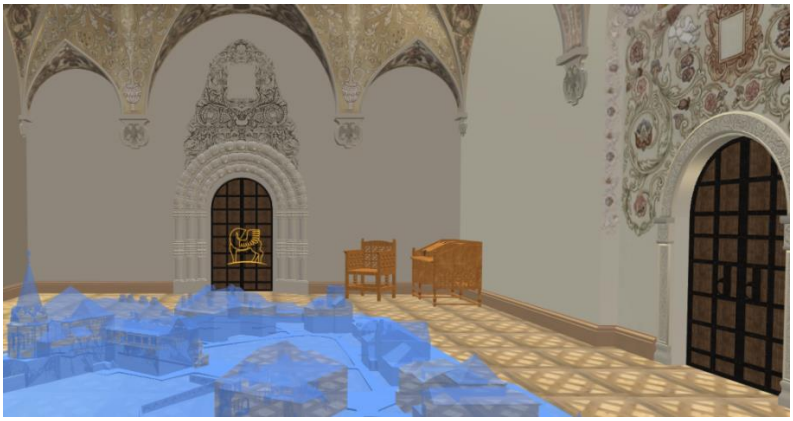


Рис. 3. Указатель спроектирован в виде Пегаса, подсвечивающегося желтым цветом при наведении на дверь, через которую можно пройти

Заключение. В дальнейшем планируется создание трехмерной модели указателя, стилистически подходящей к интерьерам Трапезной палаты, с возможностью подсвечивать разные части указателя в зависимости от того, которую из кнопок контроллера можно использовать в каждом конкретном случае; продолжается постепенное наполнение музея различными экспонатами (исторические фотографии и предметы интерьера). Также планируется создание аудиогида на русском и английском языках с возможностью переключения между языками внутри приложения. Аудиозаписи будут привязаны к перемещениям по музею и к взаимодействию с экспонатами.

Литература

1. Борисов Н.В., Лавров А.В., Мельников В.Л., Смолин А.А., Сопроненко Л.П., Захаркина В.В. Digital humanities как коммуникативное практико-ориентированное виртуальное пространство проектов цифрового культурного наследия // Информационные технологии в гуманитарных науках: сборник докладов Международной научно-практической конференции. – 2018. – С. 103–110.
2. Sopronenko L.P., Akulaeva E.A. The virtual reconstruction of the interior of Refectory Feodorovsky town in Tsarskoye Selo // Electronic Imaging & the Visual Arts. – 2015. – P. 119–121.
3. Сопроненко Л.П., Лаврова А.В., Смолин А.А., Мельников В.Л. Использование мультимедийных технологий для интерактивной трехмерной реконструкции интерьера Трапезной палаты Феодоровского городка // Культура и технологии. – 2017. – Т. 2. – № 4. – С. 89–96.

Лавская Лина Владимировна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41601
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: lavskayalv@yandex.ru

Зенцова Елена Сергеевна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № K41401
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: zencova1996@mail.ru

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004.51

АНАЛИЗ ТЕНДЕНЦИЙ ПРИМЕНЕНИЯ UI/UX-ТЕХНОЛОГИЙ К СОЗДАНИЮ ИНТЕРФЕЙСА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Лавская Л.В., Зенцова Е.С.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

В работе был проведен анализ существующих тенденций применения UI/UX-технологий к созданию интерфейса мобильного приложения. В ходе него были определены основные тенденции UI/UX-технологий к созданию интерфейса, выделены преимущества каждой из них, а также сделаны выводы по проделанному анализу.
Ключевые слова: анализ тенденций, UI, UX, интерфейс, мобильные приложения, дизайн интерфейсов.

Мобильный рынок очень динамичен и с каждым днем количество производителей программных продуктов увеличивается, поэтому необходимо учитывать всевозможные тенденции, чтобы будущее приложение оставалось конкурентоспособным [1].

Огромную роль в успехе мобильного приложения играет пользовательский интерфейс или же UserInterface (UI), занимающийся визуальным дизайном приложения. Но прежде всего, необходимо организовать информацию в приложении так, чтобы она была доступна для восприятия, т.е. адаптировать ее для взаимодействия пользователя с интерфейсом. Этим занимается технология UX (UsereXperience).

Для того чтобы создать дизайн успешного приложения необходимо объединить технологии UI и UX. Как и любая другая сфера деятельности человека технологии UI/UX продолжают развиваться.

В ходе исследования были выделены следующие тенденции UI:

- контрастные цвета и градиенты;
- большой размер шрифтов;
- анимации;
- стремление к минимализму в дизайне.

Что касается пользовательского взаимодействия или UX, то здесь наблюдаются:

- короткие формы для заполнения данных, как правило, состоящие из нескольких полей;
- стремление к более интуитивному взаимодействию пользователя с сайтом.

После определения тенденций каждой технологии отдельно были выделены общие преимущества UI/UX-дизайна.

1. Контрастные цвета. Они помогают визуально выделить область, куда пользователь должны обратить свой взор в первую очередь, будь то кнопка для получения какой-либо услуги, или же текст, который явно его заинтересует [2].
2. Большой размер шрифтов. Обычно помогает выделить основную идею сайта, или определенной страницы [2].
3. Минимализм. Минимализм в дизайн пришел вместе с популярностью мобильных устройств. С каждым годом посещение мобильных устройств или планшетов возрастает, поэтому появилась необходимость оставлять на веб-страницах только самую важную информацию, ведь размеры экранов мобильных устройств в разы меньше настольных персональных компьютеров [2].
4. Анимация. Анимация в дизайне интерфейсов мобильных приложений уже стала распространенным трендом, но с каждым годом визуальные эффекты становятся более функциональными. Анимации демонстрируют связь различных состояний интерфейса, указывают на взаимодействие между общими элементами, присутствующими в разных состояниях и привлекают внимание пользователей к тому, что они должны обязательно заметить [3].
5. Короткие формы связи. Могут значительно повысить интерес пользователя, оставить свои данные на сайте, ведь спросить по телефону, чем именно клиент заинтересовался, гораздо быстрее, чем заставлять его заполнять некоторое количество полей. Но здесь стоит учитывать бизнес-процессы компании и ее территориальную распределенность [2].
6. «Говорящие иконки». Помогают пользователю, не читая текст, понять его смысл. Также «говорящие иконки» очень распространены в мобильных версиях сайта или приложениях, заменяя собой названия пунктов меню или текст кнопок функционала.

Помимо технических нюансов существуют модные тенденции, которых следует придерживаться, чтобы приложение выглядело современным и актуальным [2].

На данный момент основная такая тенденция – Material Design от Google.

Главной его целью является создание унифицированного пользовательского пространства на всех платформах и устройствах, независимо от их размера и диагонали экрана.

Визуальными особенностями «Material» являются насыщенные, ровные цвета, резкие и очерченные края, крупная типографика и большие отступы между элементами.

На действиях пользователя сфокусировано основное внимание. Взаимодействием с дизайном управляет пользовательский опыт, а не наоборот. Все действия происходят в одном окружении, интерактивные объекты без прерывания последовательности переходят из одной среду в другую [4].

Мобильные технологии с каждым днем становятся все более популярными, поэтому разработка мобильных приложений и интерфейсов в настоящее время необходима [5].

На данный момент отсутствует конкретизированная систематизация требований к дизайну интерфейса. Это обусловлено постоянно меняющимися тенденциями в сфере дизайна.

В ходе анализа научных исследований удалось выделить основные требования и тенденции применения технологий UI/UX-дизайна к интерфейсу мобильного приложения.

1. Чем проще интерфейс, тем меньше усилий пользователь прилагает, а это хорошо сказывается на пользовательском опыте. На сегодняшний день минимализм является одним из главных тенденций. Простота цветов фона и легкий доступ к навигации создает приятное впечатление от интерфейса.
2. Цвет в дизайне интерфейса акцентирует на главном. Например, дизайнеры могут использовать цвета, чтобы установить визуальное различие между разными типами уведомлений.
3. Быстрое время загрузки. Долгое ожидание загрузки страницы отталкивает пользователя.
4. Использование анимации между страницами или объектами интерфейса.
5. Принцип «диалогового дизайна». Общение с пользователем посредством чатботов и голосовых помощников, усиленных искусственным интеллектом.

Литература

1. Дворянин Д.М., Скубников К.С. К вопросу о понятиях UX UI дизайна // Аллея науки. – 2017. – Т. 2. – № 15. – С. 781–785.
2. Логинов М.Ю. Исследование современных тенденций UI/UX // Альманах научных работ молодых ученых Университета ИТМО. – 2017. – Т. 4. – С. 63–65.
3. Шепелёв М.Д. Исследование методов проектирования UX мобильных интерфейсов: маг. диссерт. – СПб.: Университет ИТМО, 2016.
4. Material Design [Электронный ресурс]. – Режим доступа: https://material.io/ (дата обращения: 04.01.2019).
5. Мозгалева А.И., Реннит А.А. Разработка интерфейса мобильного приложения // Молодежь и современные информационные технологии: сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых. – Томск: Изд-во ТПУ, 2016. – Т. 2. – С. 18–19.

Литвинов Кирилл Владимирович
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4263
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: squertelsquertel@mail.ru

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004

ПРОБЛЕМА ВЗАИМОДЕЙСТВИЯ УСТРОЙСТВ В СФЕРЕ ИНТЕРНЕТА ВЕЩЕЙ

Литвинов К.В.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

В работе рассмотрен вопрос, касающийся взаимодействия устройств в сфере интернета вещей, при рассмотрении которого была выявлена проблема передачи данных между устройствами. Данная проблема возникает из-за вариативности аппаратных решений и протоколов передачи данных устройств, которые они используют. Также представлено решение поставленной проблемы при помощи идентификации подсетей и использования прикладного программного интерфейса.
Ключевые слова: интернет вещей, взаимодействие, глобальная сеть, масштабируемость, RESTful API.

Международный Союз Электросвязи обозначил, что ключевой проблемой интернета вещей (Internet of Things, IoT) является несовместимость протоколов и устройств, и добавил, что существует необходимость «общения» устройств между собой [1]. Все устройства, исходя из концепции IoT, должны взаимодействовать друг с другом и обмениваться определенной информацией. Устройства разрабатываются при помощи различных архитектурных решений, а также имеют свойственные себе интерфейсы и протоколы. В одной из статей журнала Платформы промышленной аналитики, Ли Да Сюй и Ву Хе отметили, что отсутствие общепринятого языка описания затрудняет развитие IoT и усложняет масштабируемость устройств.

Проблема совместимости на данный момент решается такими компаниями, как: Google, Cisco, Microsoft и др.

Исходя из вышеописанного, можно сделать вывод, что одной из проблем интернета вещей является взаимодействие устройств с окружающим миром – другими устройствами, людьми и т.д.

Сейчас практически каждое устройство в своей аппаратной части имеет встроенный Wi-Fi или ethernet модуль, при помощи которого оно способно взаимодействовать с глобальной сетью Интернет. Можно предположить, что было бы корректно, если бы эти вещи «общались» между собой через программный прикладной интерфейс для взаимодействия компонентов в сети (RESTful API). Данный метод позволит решить проблему совместимости и масштабируемости при грамотно реализованном API. В своей книге Мачей Кранц [2], а также Alexandra Bowen в цикле статей про интернет вещей [3] высказали мысль о том, что мир IoT должен быть открытым, и любое устройство должно иметь возможность взаимодействовать с другими устройствами. Исходя из этого, можно сделать вывод, что любой пользователь,

использующий интернет вещей, должен иметь возможность использовать устройства другого пользователя по договоренности.

Поставленная задача решается таким образом: всю глобальную сеть, где будут использоваться IoT, можно разбить на маленькие, более частные подсети со своим ключом от нее. Если в какой-то определенный момент пользователь X захочет предоставить доступ к своим устройствам пользователю Y, то пользователь X дает ключ от своей подсети пользователю Y. Таким образом, пользователь Y может взаимодействовать с устройствами, расположенными в сети пользователя X.

Для того чтобы глобальная сеть была разделена на подсети, где каждый является участником только своего собственного, приватного IoT-пространства, необходимо каждую такую сеть идентифицировать. Из рисунка видно, что пользователь может управлять своей подсетью, в которой размещаются устройства.

Изначально, чтобы разделить сеть на подсети, пользователю следует зарегистрировать token-ключ для своей подсети, который будет идентифицировать эту подсеть. Дальнейшее взаимодействие с этой подсетью будет происходить при помощи этого token-ключа.

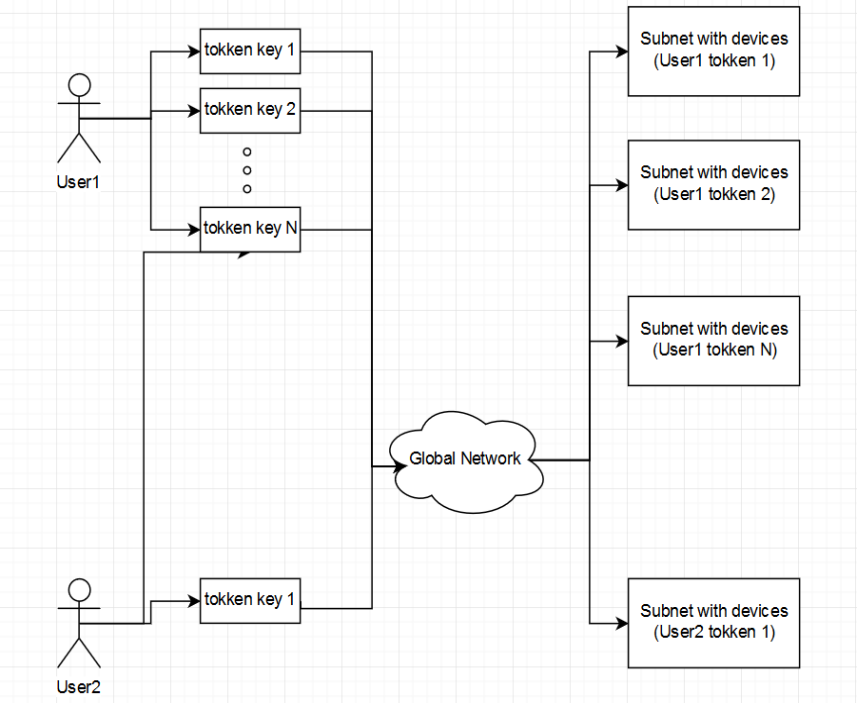


Рисунок. Общее представление концепции взаимодействия пользователей, подсетей и устройств в подсети

В работе описана проблема масштабируемости и взаимодействия устройств в сфере интернета вещей. Была предложена концепция возможного решения данной проблемы при помощи RESTful API, а также идентификации подсетей с устройствами и пользователей с помощью token-ключей.

Литература

1. Ключевые проблемы Интернета вещей – совместимость протоколов и устройств [Электронный ресурс]. – Режим доступа: https://sk.ru/news/b/articles/archive/2015/11/19/klyuchevaya-problema-interneta-vescheу-_1420_--sovmestimost-protokolov-i-ustroystv.aspx (дата обращения: 04.01.2019).
2. Кранц М. Интернет вещей. Новая технологическая революция. – М.: Бомбора, 2018. – 336 с.
3. IoT is eating the world: Current state of IoT [Электронный ресурс]. – Режим доступа: https://medium.com/@AlexandraBowen/iot-is-eating-the-world-ed97f2b5a2ca (дата обращения: 04.01.2019).

Максименко Александра Евгеньевна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4272
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: alexamaximenko@gmail.com

Смолин Артем Александрович
Год рождения: 1977
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.философ.н.
e-mail: smolin@corp.ifmo.ru

Корпан Лидия Михайловна
Год рождения: 1967
Университет ИТМО, факультет программной инженерии и компьютерной техники, ст. преподаватель
e-mail: lycor@mail.ru

УДК 004.51

ВАРИАТИВНЫЕ ШРИФТЫ КАК НОВАЯ ТЕХНОЛОГИЯ ЭКРАННОЙ ТИПОГРАФИКИ

Максименко А.Е., Смолин А.А., Корпан Л.М.
Научный руководитель – к.философ.н. Смолин А.А.

Работа выполнена в рамках темы НИР № 245922 «Исследование удобочитаемости кириллических шрифтов электронных изданий».

В работе рассмотрен новый формат OpenType 1.8, который подразумевает внедрение технологии вариативных шрифтов в пользовательские интерфейсы. Сформулированы основные достоинства и недостатки вариативных шрифтов, а также определены дальнейшие пути развития данной технологии и его влияния на экранную типографику.
Ключевые слова: вариативные шрифты, типографика, дизайн, удобочитаемость, пользовательский интерфейс.

Шрифтовой дизайн является одной из консервативных областей дизайна. Он во многом зависит от технологий и вместе с этим влияет на них. Главное, что произошло и происходит всегда – именно шрифт определяет, какие технологии и средства будут использованы, и как будет выглядеть наш визуальный мир. Известно, что вместе со шрифтом изобретались чернила, бумага, новые технологии. Как правило, это приводило лишь к упрощению воспроизводства шрифта в конкретное время и в конкретном месте. Его графическая форма никак потенциально и кардинально не эволюционировала. Рукописный шрифт упрощался до металлического набора и вместе с этим терял свой индивидуальный характер. Однако сегодня

появилась первая технология, которая позволяет не упростить шрифт, но создать что-то новое в типографике и в шрифтовом дизайне в целом.

Примерно от начала нашей эры до изобретения металлического набора (XV век), когда рукописная форма зафиксировалась и превратилась в металл, люди постоянно экспериментировали со шрифтом. Однако после того как литеры были отлиты в металле, его эволюция остановилась. Можно утверждать, что за последние 500 лет шрифты практически не изменились [1, С. 14].

Все изменилось в 2016 году, когда на конференции ATypI ведущие компании рынка Adobe, Microsoft, Apple и Google объявили о новой технологии вариативных шрифтов, которая нашла свою реализацию в усовершенствованной версии формата OpenType 1.8. Под вариативными шрифтами понимается расширение формата OpenType, благодаря которому один-единственный шрифтовой файл ведет себя как бесконечное множество начертаний [2]. Благодаря тому, что каждый символ шрифта представляет собой контурный рисунок, файл обладает небольшим весом. Параметры начертания шрифта задаются положением точек, формирующих этот контур. Двигая данные точки, возможно, например, добиться другого начертания шрифта. В таких условиях также становится возможным настраивать промежуточные параметры для этих начертаний.

Функционирование вариативных шрифтов основано на таком математическом методе как интерполяции – способе получения неизвестных промежуточных значений при известных крайних. Крайние значения – мастера – формируют так называемые интерполяционные оси: ось насыщенности, ширины, оптического размера и наклона. Использование сразу нескольких осей позволяет получить еще больше стилевых возможностей. В этом пространстве интерполяционных взаимосвязей, представленном на рисунке ниже, дизайнер может указать нужные ему значения для шрифта.

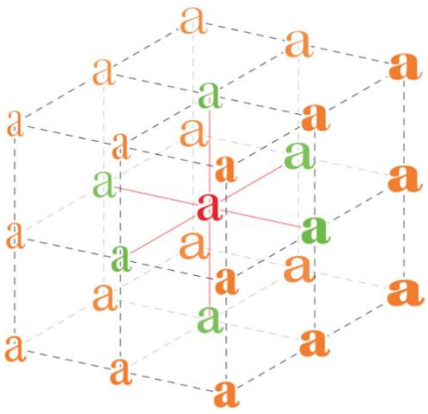


Рисунок. Шрифтовой куб Дж. Хадсона с тремя осями интерполяции: ширина, вес и оптический размер

В формате OpenType 1.8 можно выделить четыре основных вариационных осей, которыми можно управлять:

– wght – вес, который управляется свойством CSS font-weight, может принимать значения от 1 до 999;
– wdth – ширина, которая управляется свойством CSS font-stretch. Значение может представлять собой ключевое слово или процент;
– opsz – оптический размер, который может быть включен или выключен при помощи нового свойства CSS – font-optical-sizing;
– slnt – наклон, который задается при помощи установления CSS font-style наклонным oblique. По умолчанию шрифт наклонен на 20 градусов, но он также может принимать значения между –90 и 90 градусами.

Безусловно данная технология имеет ряд достоинств, способных упростить процесс работы дизайнеров и разработчиков интерфейсов [3]:

1. вариативный шрифт обеспечивает неограниченный набор начертаний. Наличие такого множества различных стилей при работе над интерфейсами позволит придать им неповторимый и яркий графический облик;
2. вариативный шрифт позволяет дизайнеру самостоятельно выбирать то или иное начертание в зависимости от конкретного проекта. Известно, каждый шрифт имеет определенное количество начертаний: light, regular, bold и так далее. Однако стоит отметить, что, например, самое жирное начертание одного шрифта может отличаться от самого жирного начертания другого шрифта. Начертания представляют собой ориентиры, которые указывает сам разработчик шрифта. Именно поэтому одни и те же начертания могут выглядеть по-разному в разных шрифтах. При использовании же вариативных шрифтов дизайнер сможет собственноручно указывать те значения начертаний, которые ему необходимы;
3. внедрение вариативных шрифтов обеспечивает отзывчивость экранной типографики. Как было отмечено ранее, вариативный шрифт имеет два крайних начертания – самое узкое и самое широкое. В связи с этим электронный шрифт способен адаптироваться под любой размер экрана: он может растягиваться или сжиматься без потери качества своего внешнего вида и отвечать всем эстетическим требованиям;
4. вариативный шрифт способен повысить уровень удобочитаемости. Удобочитаемость – это свойство, определяющее легкость и комфорт, с которым текст воспринимается и понимается [4, С. 111]. Лучшая удобочитаемость электронного текста достигается засчет изменения оптического размера. Например, при небольшом кегле шрифта его штрихи могут быть толще, а значит, более удобочитаемыми. И, наоборот, при большом размере шрифта его рисунок может быть более контрастным и четким;
5. вариативный шрифт также способен обеспечить высокую скорость загрузки веб-сайтов на уровне сервера, так как он предполагает использование всего одного шрифтового файла небольшого размера.

Однако стоит отметить некоторые трудности, которые может вызывать данная технология:

1. в вариативных шрифтах отсутствует ось курсива, в связи с чем необходимо иметь дополнительный файл для курсивного начертания. Отсутствие данной оси обусловлено тем, что курсивные и римские стили шрифтов имеют различные строения, а значит, точечные контуры символов не могут быть совместимы. Поэтому вариативные шрифты в большинстве случаев имеют два файла: для нормального и курсивного начертания;
2. вариативные шрифты предлагают большое количество возможностей изменения шрифтов, что, безусловно, является технологическим достижением. Однако вместе с этим и усложняется процесс выбора начертания. Обладая такими большими возможностями, испортить дизайн интерфейса становится гораздо проще. Решение данной проблемы видится в повышении квалификации дизайнера и разработчика и получении более углубленных знаний в области типографики.

Сегодня вариативные шрифты поддерживают более 65% браузеров, а также графические редакторы Adobe Photoshop и Adobe Illustrator [5], что свидетельствует о том, что данная технология вполне применяема. Бесспорным является и тот факт, что изобретение вариативных шрифтов можно считать настоящим прорывом, благодаря которому открывается целый ряд направлений для развития типографики как в производственной сфере, так и в пользовательской. Использование вариативных шрифтов, прежде всего, делает возможным обширное внедрение технологий улучшения восприятия информации в интерфейсы, начиная с этапа их проектирования и вплоть до удобства использования. Среди главных преимуществ технологии: и оптимизация объема данных, и улучшенная адаптивность типографики, и повышенная удобочитаемость контента. Данная технология также открывает новые перспективы в экранной типографике – функции автоматической подстройки шрифта под пользователя в зависимости от его предпочтений или взаимосвязей шрифта со смысловой или

эмоциональной нагрузкой контента. Главное, что изменится с внедрением вариативных шрифтов – у пользователей появится вкус чтения. Данная технология позволяет сделать шрифт вновь разнообразным, эмоциональным и индивидуальным, а значит, наши слова смогут обрести лучший типографский голос.

Литература

1. Филиппович А.Ю. Разработка шрифтовых элементов. Курс практических занятий. – М.: ПЦ, 2008. – 129 с.
2. Рудерман И. Знакомьтесь, вариативные шрифты (variable fonts)! [Электронный ресурс]. – Режим доступа: https://type.today/ru/journal/opentype_variables (дата обращения: 15.01.2019).
3. Schöndorfer O. How to start with variable fonts on the web [Электронный ресурс]. – Режим доступа: https://www.zeichenschatz.net/typografie/how-to-start-with-variable-fonts-on-the-web.html (дата обращения: 15.01.2019).
4. Феличи Дж. Типографика: шрифт, верстка, дизайн / Пер. с англ. – 2-е изд. – СПб.: БХВ-Петербург, 2014. – 496 с.
5. Rutter R. How to use variable fonts in the real world [Электронный ресурс]. – Режим доступа: https://clearleft.com/posts/how-to-use-variable-fonts-in-the-real-world (дата обращения: 15.01.2019).

Мариничев Никита Андреевич
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P4270
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: nikitus.1995@yandex.ru

Меженин Александр Владимирович
Год рождения: 1959
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: mejenin@mail.ru

УДК 004.92

ОЦЕНКА КАЧЕСТВА ЗАХВАТА ДВИЖЕНИЯ ПРИ ИСПОЛЬЗОВАНИИ СВЯЗКИ ИЗ ДВУХ И ТРЕХ MICROSOFT KINECT 2.0

Мариничев Н.А., Меженин А.В.
Научный руководитель – к.т.н., доцент Меженин А.В.

В работе рассмотрены вопросы оценки качества захвата движений системами Kinect. Исследован вопрос, касающийся целесообразности использования трех модулей Kinect в сравнении с использованием двух модулей. Для оценки предложено использовать значения угловых ускорений узлов кости между соседними кадрами.
Ключевые слова: системы захвата движений, оценка качества захвата движений.

Захват движений используется повсеместно как в киноиндустрии, так и при разработке компьютерных игр. На данный момент существует множество различных систем захвата движений, среди которых системы компьютерного зрения на основе камер глубины являются наиболее доступным решением в соотношении цена-качество. Использование одного модуля не сможет обеспечить захват движений актера со всех сторон, однако, увеличение числа модулей позволит решить эту проблему, но потребует дополнительных вычислительных и финансовых затрат [1–5].

Для обоснованной оценки различий между использованием схем, состоящих из двух и трех модулей, необходимо для начала записать видео с каждого из модулей. Полученные при помощи программы iPi Recorder видеозаписи объединяются в один файл при помощи программы iPi MoCap Studio. На рисунке представлены схемы расстановки модулей таким образом, чтобы исследовать значения угловых ускорений одних и те же захваченных движений актера.

В рамках эксперимента были получены видеозаписи, содержащие 500 кадров. Без учета костей пальцев, не предусмотренных к захвату движения средствами Kinect, а также без учета костей, в которых не наблюдалось изменение углов поворота, были получены данные по 15 костям. Значения углов поворота Rotate по осям X, Y и Z для каждой из кости в каждом из кадров выгружались из программы iPi MoCap Studio для дальнейшей обработки в программе MS Excel.

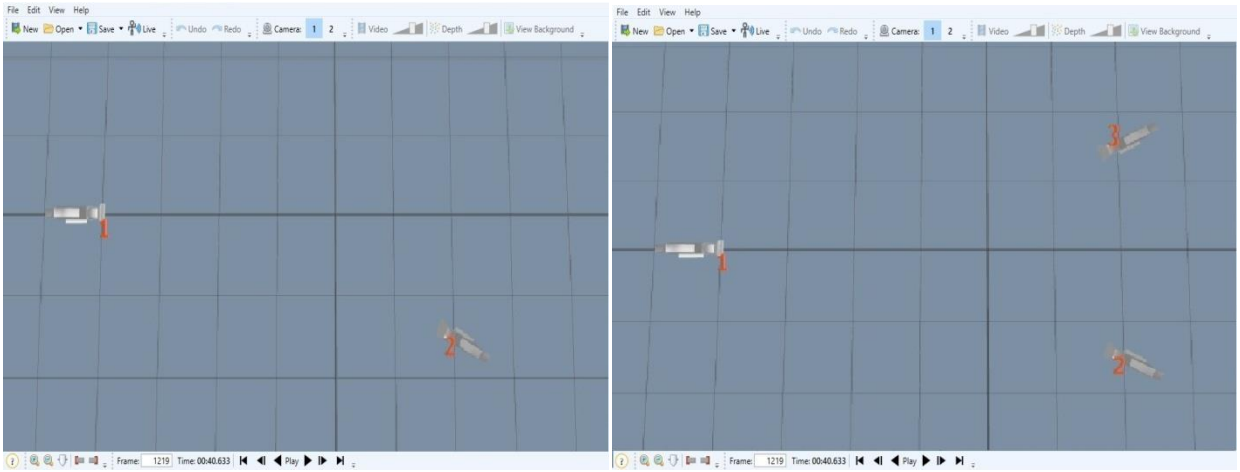


Рисунок. Схемы подключения модулей Kinect

На основании значений об углах поворота необходимо получить угловую скорость по оси *OX*, которая считается как первая производная от перемещения вдоль оси O*X*:

$$vx_{i,i+1} = \frac{x_{i+1} - x_i}{f} \text{ град/с},$$

где f =1/30 с, поскольку Kinect 2.0 имеет частоту записи видео 30 кадров/с. Угловая скорость по осям O*Y* и O*Z* считается аналогично.

Затем подсчитывается угловое ускорение по оси O*X* как вторая производная от перемещения вдоль оси O*X*:

$$ax_{i,i+1} = \frac{vx_{i,i+1} - vx_{i+1,i+2}}{f} \text{ град/с}^2.$$

Угловое ускорение по осям O*Y* и O*Z* считается аналогично.

Итоговое угловое ускорение считается, как длина вектора, образуемого ускорениями по осям O*X*, O*Y* и O*Z*:

$$a_{i,i+1} = \sqrt{(ax_{i,i+1})^2 + (ay_{i,i+1})^2 + (az_{i,i+1})^2} \text{ град/с}^2.$$

Используя вышеперечисленные преобразования, было получено две выборки значений угловых ускорений, которые в дальнейшем используются для проведения статистического анализа на основе расчета парного *t*-критерия Стьюдента. Проверяемая альтернативная гипотеза гласит, что с большей долей вероятности использование трех модулей Kinect 2.0 позволит добиться лучшего качества, т.е. будет заметно уменьшение значений угловых ускорений в сравнении с использованием двух модулей, но понесет за собой дополнительные вычислительные и финансовые затраты. Нулевая – использование трех модулей Kinect 2.0 не даст ощутимой прибавки в качестве получаемой анимации в сравнении с двумя модулями.

Итогом подсчетов служит таблица с подсчитанными значениями парного *t*-критерия Стьюдента.

Таблица. Результат подсчета парного *t*-критерия Стьюдента

кость	Hip	Middle Spine	Chest	Lclavicle	Lshoulder	Lforearm	Rclavicle	Rshoulder
t-критерий Стьюдента	-0,70736681	-0,62235784	-0,53507	3,2665561	1,1167184	2,45099346	0,333106	-0,42494
ср. арифм. Разностей	-2865,50398	-3,62101715	-10,3904	11,54898	45,07392	117,222601	1,380294	-17,0238
значимость	23,983%	26,700%	29,642%	99,942%	86,767%	99,271%	63,040%	33,553%
кость	Rforearm	Rthigh	Rshin	Rfoot	Lthigh	Lshin	Lfoot	
t-критерий Стьюдента	0,199457501	-2,00804636	-4,323082	-2,4504707	3,7512239	1,53092383	-4,77107	
ср. арифм. Разностей	11,89420566	-106,874317	-219,3132	-381,18612	194,28923	87,6233001	-220,591	
значимость	57,901%	2,259%	0,001%	0,730%	99,990%	93,679%	0,0001%	

Поскольку в данной работе рассматривались 15 нулевых гипотез, т.е. присутствовала множественная проверка гипотез, было принято решение проводить проверку каждой из гипотез на уровне значимости 0,34%. Тем самым, при проверке сразу 15 нулевых гипотез

вероятность того, что, хотя бы одна из них окажется отклоненной ошибочно, будет составлять 4,98%.

В итоге было обнаружено, что в костях Rshin и Lfoot наблюдается значимый результат на уровне значимости 0,34%, что дает основание принять альтернативную гипотезу. Однако в костях Lclavicle и Lthigh также наблюдается значимый на уровне 0,34% результат, но свидетельствующий о противоположном, т.е. о том, что угловые ускорения при использовании двух Kinect будут меньше, чем при использовании трех Kinect.

Таким образом, на данный момент нет возможности однозначно утверждать, что использование трех Kinect обеспечит более качественный захват движений, чем использование двух Kinect. В связи с этим необходимо проведение дополнительных экспериментов с использованием различных схем подключения и в условиях, более приемлемых для захвата движения. Это будет необходимо, чтобы минимизировать влияние отраженного от различных предметов используемого помещения инфракрасного излучения, создающего предпосылки к появлению нежелательного облака точек, которые могут быть причиной полученного результата с точки зрения наличия противоположных альтернативной гипотезе утверждений.

Литература

1. Егорова Е.С., Лавров А.В., Меженин А.В. Оценка точности захвата движений в рабочем пространстве системы Kinect // Наука и образование в жизни современного общества: Сборник научных трудов по материалам Международной научно-практической конференции. – 2015. – Ч. 12. – С. 31–33.
2. Коротких А.Е., Меженин А.В. Методы трекинга камеры в системах on-set превизуализации // Сборник тезисов докладов конгресса молодых ученых. Электронное издание [Электронный ресурс]. – Режим доступа: http://openbooks.ifmo.ru/ru/file/7272/7272.pdf (дата обращения: 04.01.2019).
3. Извозчикова В.В., Меженин А.В. 3D-моделирование методов съемки мобильными видеосистемами // Программные продукты и системы. – 2016. – № 3. – С. 163–167.
4. Меженин А.В. Виртуальные 3D среды как средство верификации и тестирования при проектировании // Приоритетные научные направления: от теории к практике. – 2016. – № 21. – С. 105–110.
5. Sizikov V.S., Stepanov A.V., Mezhenin A.V., Burlov D.I., Eksemplyarov R.A. Determining image-distortion parameters by spectral means when processing pictures of the earth’s surface obtained from satellites and aircraft // Journal of Optical Technology. – 2018. – V. 85. – № 4. – P. 203–210.

Машина Екатерина Алексеевна
Год рождения: 2000
Университет ИТМО, факультет программной инженерии
и компьютерных технологий, студент группы № P3110
Направление подготовки: 09.03.04 – Программная инженерия
e-mail: mashina.katherina@gmail.com

Сушенцова Марина Вячеславовна
Год рождения: 2001
Московский государственный университет им. М.В. Ломоносова,
факультет почвоведения, студент группы № 103
e-mail: vfhbif2001@gmail.com

Нелепко Любовь Николаевна
Год рождения: 1999
Санкт-Петербургский государственный университет,
факультет лечебного дела, студент группы № 102
Направление подготовки: 31.05.01 – Лечебное дело
e-mail: lykila@gmail.com

Баркалов Максим Максимович
Год рождения: 1998
Университет ИТМО, факультет информационных технологий
и программирования, студент группы № M3305
Направление подготовки: 09.03.02 – Информационные системы
и технологии
e-mail: maksimbark@corp.ifmo.ru

Бейлин Максим Тимурович
Год рождения: 1998
Университет ИТМО, факультет инфокоммуникационных технологий,
студент группы № K3320
Направление подготовки: 11.03.02 – Инфокоммуникационные
технологии и системы связи
e-mail: icevenom98@gmail.com

Орлова Любовь Дмитриевна
Год рождения: 2000
Университет ИТМО, факультет инфокоммуникационных технологий,
студент группы № K3142
Направление подготовки: 31.05.01 – Информационные технологии
в гуманитарной сфере
e-mail: epicyonhaydeny@gmail.com

Григорьева Екатерина Игоревна
Год рождения: 2000
Университет ИТМО, факультет инфокоммуникационных технологий, студент группы № К3141
Направление подготовки: 09.03.03 – Прикладная информатика
e-mail: grig.katrin@gmail.com

УДК 004.91

СИСТЕМАТИЗАЦИЯ НАЧАЛЬНОГО ИНФОРМАЦИОННО-ТЕХНОЛОГИЧЕСКОГО КОНТЕНТА, НЕОБХОДИМОГО ДЛЯ СОЗДАНИЯ СИСТЕМЫ УДАЛЕННОЙ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ МУНИЦИПАЛЬНЫХ ДРУЖИН ДОБРОВОЛЬНОЙ САМОПОМОЩИ PERSONALMEDHELPER

Машина Е.А. (Университет ИТМО)**, Сушенцова М.В.** (Московский государственный университет им. М.В. Ломоносова)**, Нелепко Л.Н.** (Санкт-Петербургский государственный университет)**, Баркалов М.М.** (Университет ИТМО)**, Бейлин М.Т.** (Университет ИТМО)**, Орлова Л.Д.** (Университет ИТМО)**, Григорьева Е.И.** (Университет ИТМО)
Научный руководитель – к.т.н. Балакшин П.В. (Университет ИТМО)

Работа выполнена в рамках инициативных разработок по проекту PersonalMedHelper.beginning осуществляемому за счет гранта ITMOSTUDENTS.Initiative.

В работе рассмотрены результаты работ совместной межотраслевой инициативной группы студентов Университета ИТМО, Московского государственного университета им. М.В. Ломоносова и Санкт-Петербургского государственного университета начального этапа создания семейства пациенто-ориентированных телемедицинских систем семейства PersonalMedHelper. Проанализированы особенности подходов к конструированию подобных систем на примере создания системы удаленной информационной поддержки добровольных муниципальных дружин.
Ключевые слова: информационная система, телемедицина, открытые системы, большие данные, медицинские добровольцы, первичная медико-санитарная помощь, социально-экономический эффект.

На сегодняшний день телемедицина является одним из наиболее динамично развивающихся рынков [1]. Исследовательское агентство BBC Research прогнозирует к 2019 году объемы глобального рынка телемедицины около 40 млрд долл. при среднегодовом росте в размере 20% [2].

Причем развитие телемедицинских технологий в разных странах происходит с разной скоростью. Так, во втором квартале 2018 года искусственный интеллект стал одним из наиболее прорывных направлений для инвестиций в США с объемом финансирования 2,3 млрд долл. [2]. Россия же серьезно отстает от США и Китая по инвестициям в развитии информационных технологий в различных отраслях медицины [3], однако, востребованность телемедицинских решений, в том числе и основанных на элементах искусственного интеллекта, в нашей стране растет [4].

Дополнительным фактором, повышающим востребованность телемедицинских технологий, являются уникальные географические условия нашей страны, большие расстояния, высокая концентрация населения в европейской части и небольшая плотность проживания за Уралом [5].

Несмотря на все возрастающий дефицит современных отечественных телемедицинских решений, ощущаемых практически во всех сферах отечественного здравоохранения [6], наиболее остро стоит вопрос создания системы информационной поддержки муниципальных медицинских волонтеров, занимающихся оказанием доврачебной медицинской помощи

населению. Волонтеры осуществляют свою деятельность в соответствии с Приказом Минздравсоцразвития России от 15.05.2012 № 543н [7]. Несмотря на то, что единое решение, призванное обеспечить всей необходимой информацией медицинских волонтеров на местах, крайне востребовано [6], подобные системы пока не находятся в свободном доступе у парамедицинских волонтеров [8].

В связи с этим консорциум инициативных студенческих групп Московского государственного университета им. М.В. Ломоносова (МГУ), Санкт-Петербургского государственного университета (СПбГУ) и Университета ИТМО поставил перед собой цель: создание системы информационной поддержки муниципальных медицинских волонтеров, занимающихся оказанием доврачебной медицинской помощи населению, получившей название PersonalMedHelper (PMH).

В связи с тем, что PMH изначально создается как решение, предполагающее постоянное расширение своего функционала, то его разработка основывается на принципах создания открытых систем [9] и ведется в публичном пространстве коворкинга PMH.Open, предполагающем привлечение к работам не только студентов указанных университетов, но и сторонних участников, работающих, как по непосредственным заданиям PHM.Team, так и по собственной инициативе.

В своей работе участники PMH.Team в качестве информационно-технологического задела используют собственные предварительные разработки, которые с минимальными изменениями стали основой для выполнения работ по проекту PMH.

Со стороны ITMO-PMH.Team такой базой явилась цифровая платформа поддержки волонтерских инициатив EcoDataKeeper, про которую создатели демонстрировали на VII Конгрессе Молодых Ученых ИТМО [10].

Работа команды МГУ-PMH.Team базируется на собственных результатах исследовательских работ по обоснованию социально-экономического эффекта от создания пациенто-ориентированных телемедицинских систем, выполненных в рамках сотрудничества с Русским географическим обществом. Группа СПбГУ использует в проекте результаты собственных инициативных работ по начальной систематизации парамедицинского контента.

Согласно результатам предшествующих работ, проведенных PMH.Team работ [10–12], были сформулированы основные направления исследований, проведение которых должно предшествовать созданию решения PersonalMedHelper.

В связи с этим на настоящий момент межуниверситетская мультидисциплинарная команда PMH.Team, работающая над созданием системы информационной поддержки муниципальных медицинских волонтеров, занимающихся оказанием доврачебной медицинской помощи населению, выполняет работы по трем основным направлениям: информационные технологии, лечебное дело, правоведение.

Создание телемедицинских информационных систем является сложной и дорогостоящей задачей, подверженной большому количеству рисков [13, 14]. В связи с этим для обеспечения эффективной организации работ в случаях высокой неопределенности конечного состава продукта применяется спиральная модель разработки [15], основанная на классическом цикле Деминга. В спиральной модели разработки определен общий набор контрольных точек, первой из которой является разработка концепции эксплуатации системы (CONOPS), предназначенной для пользователя и содержащей информацию о рабочих характеристиках системы с точки зрения пользователя, описывающей один или несколько взаимосвязанных сценариев эксплуатации системы [9].

При формировании содержания, полноты и структуры профессионального медицинского контента, предоставляемого по запросу, необходимо правильно определить основную группу пользователей системы. Для определения «профессионально-квалификационного» уровня пользователя системы PMH были всесторонне изучены лучшие практики создания парамедицинского контента [8, 16, 17] и существующей отечественной законодательной базы, регламентирующей порядок оказания первичной медико-санитарной

помощи взрослому населению [7] было принято решение ориентироваться на «типичного представителя домохозяйства», т.е. гражданина, имеющего свидетельство о среднем образовании и не обладающего профессиональными медицинскими знаниями и навыками [16].

В качестве исходных данных для анализа территориальной востребованности парамедицинских услуг, оказываемых добровольцами, используются данные из общедоступных источников и типовые методы расчетов востребованности медицинских услуг, используемые при планировании ресурсов, задействованных для определения требующихся населению медицинских услуг.

При формировании лицензирующих правил, описывающих алгоритмы, ограничения и ответственности, порождаемые использованием информационной системы поддержки муниципальных медицинских волонтеров РМН, использовались результаты анализа массивов правовой информации, регламентирующих самодеятельное оказание первичной медико-санитарной помощи взрослому населению в малочисленных населенных пунктах с числом жителей менее 100 человек, в том числе временных (сезонных), находящихся на значительном удалении от медицинских организаций или их структурных подразделений [18–21].

Одним из самых важных результатов текущего этапа работ по созданию пациенто-центричной системы информационной поддержки медицинских волонтеров РМН является создание механизмов открытого пространства коворкинга РМН.Open, позволяющего одновременно, и основываясь на единственном наборе исходных данных, вести разработку продукта РМН по всем направлениям, включая обсуждение алгоритмов, поэтапного программирования, создания специализированного медицинского контента, ориентированного на непрофессионального пользователя, и создание документов, регламентирующих использование продукта РМН.

Разработанное для проекта РМН открытое пространство разработки РМН.Open содержит большое количество преднастроенных шаблонов отчетных форм, workflow, поисковых запросов и процедур хранения промежуточной информации. Все это позволяет вести создание, утверждение и корректировку обрабатываемых проектных данных большому количеству пользователей при обязательном учете внесенных комментариев и изменений во все связанные документы, вести истории изменений и управлять версиями.

Эти возможности позволяют не только существенно ускорить процесс разработки элементов конечного продукта, и кратно сократить количество возникающих ошибок, но и планировать работы, и контролировать их выполнение. Все это позволяет говорить о том, что РМН.Open может рассматриваться в качестве самостоятельного продукта – среды создания мультидисциплинарных решений.

Еще одним текущим промежуточным результатом работ является создание концепции эксплуатации системы РМН.CONOPS, предназначенной для пользователя РМН и содержащей информацию о рабочих характеристиках системы информационной поддержки муниципальных медицинских волонтеров с точки зрения пользователя, описывающая один или несколько взаимосвязанных сценариев эксплуатации системы. Отличительной особенностью созданной РМН.CONOPS является то, что она представляет собой суперпозицию базового (PMH.CONOPS.basic) и специализированного (PMH.CONOPS.medvolunteer) компонентов. Подобное двухуровневое решение позволяет использовать базовый компонент (PMH.CONOPS.basic) в качестве основы для реализации других решений РМН-семейства, которые уже имеются в текущих планах разработки межуниверситетской рабочей группы РМН.Team.

Продемонстрированный в работе модульный принцип к формированию конечного решения использован РМН.Team и при проведении работ по другим профессиональным специализациям.

Так, при создании специализированного гармонизированного информационно-справочного медицинского контента, ориентированного в рамках создания системы

информационной поддержки муниципальных медицинских волонтеров, любой контент-юнит проектируется состоящим из базового (PMH.DBaseUnit) и специализированного (PMH.DSpecUnit) компонентов, позволяющих выделить общемедицинскую (PMH.DBaseUnit) (а значит, применимую и в других телемедицинских продуктах PMH-семейства) часть и часть медицинского алгоритма, непосредственно ориентированного на медицинского волонтера, оказывающего доврачебную помощь.

Разработанный на текущем этапе создания системы PMH принцип модульности построения решений продуктов PMH-семейства позволяет существенно облегчить привлечение к работам по созданию решений независимых групп сторонних разработчиков путем опубликования в общедоступном пространстве коворкинга PMH.Open открытых описаний базовых и специализированных CONOPS и контент-компонентов. Подобная дефрагментация, по мнению PMH.Team, позволит существенно ограничить объем задач каждого из сторонних разработчиков, соответственно, снижая их трудозатраты на проверку и отладку результатов, а также повышая качество и скорость их работы.

Еще одним преимуществом модульности является продемонстрированная в ходе работ возможность построения на его основе системы регулирования пользовательского контента, исходя из медицинской квалификации пользователя. Причем, если начальный уровень предоставляемой PMH информации рассчитан на квалификацию неподготовленного волонтера (а также пациента и его близких), то при наличии у потребителя информации, о подтвержденной медицинской квалификации, объем предоставляемых по запросу данных и сервисов может быть регулируемо увеличен путем предоставления дополнительных расширений (PMH.DSpecUnit.adv).

Следующим шагом развития исследовательских работ в области модульной сегментации специализированного медицинского контента будут работы по автоматизированному определению квалификации пользователя системы путем анализа тезауруса, принадлежащих теме запросов. Это, в свою очередь, позволит существенно увеличить объем доступной для него информации, понимание которой будет соответствовать его медицинской квалификации.

Анализ результатов проведенных работ по созданию системы информационной поддержки позволил сделать целый ряд выводов.

Применение открытого пространства коворкинга PMH.Open (представляющего специально разработанное программное решение по управлению проектными данными):

- существенно упрощает и ускоряет процесс разработки сложных междисциплинарных продуктов (подобных системе информационной поддержки медицинских волонтеров PMH);
- облегчает подключение к процессу мультидисциплинарной разработки сторонних рабочих групп различных специализаций;
- позволяет осуществлять гибкое управление ходом процесса выполнения работ по проекту, фрагментируя «критические задачи» и используя параллельную разработку;
- позволяет считать PMH.Open самостоятельным продуктом, предназначенным для ведения мультидисциплинарных проектов.

Использование принципа дефрагментации всех выполняемых задач при создании системы PMH на базовый (base) и специализированный (spec) компоненты позволяет:

- осуществить наследуемый модульный принцип построения решений PMH-семейства, используя накопленный опыт предыдущих разработок;
- обеспечить возможность дальнейшего развития решений PMH-семейства в сторону создания глобальной телемедицинской системы поддержки диагностических и лечебных решений;
- создать механизм автоматизированной кастомизации контента, доступного пользователю исходя из анализа тезауруса и семантики его запросов, позволяющий осуществлять разнонаправленные трансферы глубины пользовательского контента от медико-

профессионального до пациенто-ориентированного уровня, увеличивая число потенциальных пользователей решения.

Все перечисленное говорит о том, что в процессе текущего этапа создания системы поддержки муниципальных медицинских волонтеров не только проведены работы, предшествующие созданию макетного образца системы РМН, но и получены результаты, позволяющие говорить о том, что межуниверситетская инициативная рабочая группа РМН.Team создала серьезную основу для расширения тематики работ в сторону создания масштабируемой серии телемедицинских продуктов РМН-семейства.

Представленные в настоящей работе материалы позволяют говорить о том, что создание мультидисциплинарных межуниверситетских рабочих групп, подобных РМН.Team позволяет не только значительно расширить суммарные компетенции участников, но и добиться серьезного мультипликативного эффекта, еще более повышающего качество создаваемого решения.

В связи с этим можно говорить о том, что работа по созданию системы поддержки муниципальных медицинских добровольцев РМН будет успешно завершена, а полученные на настоящий момент теоретические результаты будут использованы для создания широкой линейки телемедицинских продуктов РМН-семейства.

Литература

1. Андрющенко Г.И., Потехина Е.В. Экономические предпосылки развития рынка услуг здравоохранения в Российской Федерации // Социальная политика и социология. – 2015. – Т.14. – № 5. – С. 6–14.
2. Перспективы телемедицины: выживут крупные и узкопрофильные IT-компании [Электронный ресурс]. – Режим доступа: https://telemedicina.ru/news/russian/perspektivy-telemediciny-vyzhivut-krupnye-i-uzkoprofilnye-it-kompanii (дата обращения: 01.11.2018).
3. Остапенко К.А., Пастушкова П.С. Формирование профессиональных компетенций на основе здоровьесберегающих технологий // Аспирант и соискатель. – 2015. – № 3. – С. 66–68.
4. Сисигина Н. Развитие инфраструктуры здравоохранения в 2014–2016 гг. // Экономическое развитие России. – 2018. – № 5. – С. 68–72.
5. Щепин В.О. Государственная и муниципальная системы здравоохранения Российской Федерации в 2013 году: обеспеченность основными ресурсами, объемы и показатели деятельности, региональные особенности // История науки и техники. – 2014. – № 11. – С. 49–62.
6. Костенко А.Н., Свитайло Д.Н., Смиянов А.В. Оценка доступности медицинской помощи населению: опыт регионального исследования // Социологические исследования. – 2013. № 12. – С. 133–136.
7. Приказ Минздравсоцразвития России от 15.05.2012 № 543н «Об утверждении Положения об организации оказания первичной медико-санитарной помощи взрослому населению» (зарегистрировано в Минюсте России 27.06.2012 № 24726).
8. Шевлякова В.А. Управление процессом капитализации социального здоровья в условиях городской среды // Социальное здоровье: проблемы и решения. – 2018. – № 2. – С. 5–17.
9. Мирошниченко Е.А. Технологии программирования. – 2-е изд., испр. и доп. – Томск: Изд-во Томского политехнического университета, 2008. – 128 с.
10. Лисовский Д. Победительница конкурса ITMO.STARS Екатерина Машина: Я уверена, что по-настоящему мир меняют лишь почти неисполнимые мечты [Электронный ресурс]. – Режим доступа: http://news.ifmo.ru/ru/startups_and_business/startup/news/7940/ (дата обращения: 07.11.2018).
11. Блинникова Н. С наукой до вуза: какие исследования проводят школьники для Конгресса молодых ученых [Электронный ресурс]. – Режим доступа: http://news.ifmo.ru/ru/education/trend/news/7483/ (дата обращения: 01.11.2018).

12. Меньшикова Е. Победители конкурса ITMO.STARS – о самом неклассическом поступлении в вуз, ЕГЭ и о том, как сделать собственный проект еще в школе [Электронный ресурс]. – Режим доступа: http://news.ifmo.ru/ru/education/students/news/7847/ (дата обращения: 07.11.2018).
13. Руденко М.Н., Гаухберг Л.Д. К вопросу об оценке эффективности системы здравоохранения // Национальные интересы: приоритеты и безопасность. – 2012. – № 17. – С. 57–64.
14. Григорьева Н.С. Методологические подходы к разработке концепции развития социальной сферы (на примере здравоохранения) // Вестн. Моск. Ун-та. Сер.6. Экономика. – 2009. – № 4. – С. 3–27.
15. Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. – М.: РИА «Стандарты и качество», 2008. – 408 с.
16. Черкашина Т.Ю. Дифференциация населения России в получении медицинских услуг: значение индивидуальных и территориальных факторов // Регион: экономика и социология. – 2014. – № 11. – С. 162–182.
17. Беличенко С.А. Высокие технологии в современной медицине // ЭКО. – 2011. – № 10. – С. 48–58.
18. Наумов В.Б., Савельев Д.А. Правовые аспекты телемедицины. – СПб.: Анатолия, 2002. – 107 с.
19. Штыкова Н.Н. Сущность и проблемы реализации электронной медицины (на примере Владимирской области) // Медицинское право. – 2014. – № 5. – С. 22–27.
20. Федеральный закон от 21.11.2011 № 323-ФЗ «Об основах охраны здоровья граждан в Российской Федерации» // СЗ РФ. 2011. № 48. Ст. 6724.
21. Савельев А.И. Проблемы применения законодательства о персональных данных в эпоху «Больших данных» (Big Data) // Право. Журнал Высшей школы экономики. – 2015. – № 1. – С. 43–67.

Машина Екатерина Алексеевна
Год рождения: 2000
Университет ИТМО, факультет программной инженерии
и компьютерных технологий, студент группы № P3110
Направление подготовки: 09.03.04 – Программная инженерия
e-mail: mashina.katherina@gmail.com

Сушенцова Марина Вячеславовна
Год рождения: 2001
Московский государственный университет им. М.В. Ломоносова,
факультет почвоведения, студент группы № 103
e-mail: vfhbif2001@gmail.com

УДК 004.91

ПРИМЕНЕНИЕ СИСТЕМЫ УДАЛЕННОЙ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ ДОБРОВОЛЬНЫХ МУНИЦИПАЛЬНЫХ ДРУЖИН ДОВРАЧЕБНОЙ САМОПОМОЩИ PERSONALMEDHELPER ДЛЯ ПОВЫШЕНИЯ КАЧЕСТВА ЖИЗНИ ЖИТЕЛЕЙ НАСЕЛЕННЫХ ПУНКТОВ В РАМКАХ ПРОГРАММЫ УСТОЙЧИВОГО РАЗВИТИЯ ТЕРРИТОРИЙ

Машина Е.А. (Университет ИТМО)**, Сушенцова М.В.** (Московский государственный университет им. М.В. Ломоносова)
Научный руководитель – к.т.н. Балакшин П.В. (Университет ИТМО)

Работа выполнена в рамках инициативных разработок по проекту PersonalMedHelper.beginning осуществляемому за счет гранта ITMOSTUDENTS.Initiative.

В работе рассмотрены вопросы, связанные с путями оценки эффективности внедрения телемедицинских технологий путем более полного учета отложенного эффекта от их использования в рамках программы устойчивого развития территорий.
Ключевые слова: информационная система, телемедицина, устойчивое развитие, медицинские добровольцы, первичная медико-санитарная помощь, социально-экономический эффект.

Быстро развивающиеся телемедицинские системы сегодня оказывают большое влияние на повышение доступности высокотехнологичной медицинской помощи, а следовательно, и обеспечивают рост качества жизни населения [1]. Однако сознание подобных систем требует существенных финансовых затрат и не должно являться самоцелью развития технологий [2].

В связи с этим возникает вопрос об оценке экономической эффективности создания и внедрения подобных технологий в медицинскую практику [3]. Помимо этого, привлечение внешнего финансирования в обязательном порядке также требует наличия расчетов экономической эффективности решения.

Однако расчет экономической эффективности информационных медицинских систем не так прост, как кажется с первого взгляда. И дело даже не в том, что подчас очень трудно в денежном выражении оценить качественные результаты повышения качества жизни, связанные с улучшением доступности высокотехнологичных медицинских услуг.

Проблема усугубляется тем, что большая часть эффекта, достигаемого развитием телемедицины, имеет отложенный эффект и полностью реализуется не только в течение

достаточно длительных промежутков времени, но и через влияние факторов, напрямую не связанных с уровнем развития здравоохранения [4].

Изучению возможных путей учета отложенного эффекта от внедрения телемедицинских технологий и была посвящена данная работа.

Существующий уровень развития индустриальных и гуманитарных технологий формирует совершенно новые возможности для развития удаленных территорий Российской Федерации [5]. Одной из таких показательных и быстро развивающихся отраслей знания и технологий является разработка телемедицинских решений, требующая большого количества предварительных разработок в области медицины, вычислительной техники, юриспруденции, государственного управления, гносеологии и тому подобных областей [1].

Географические условия России, связанные с большими расстояниями, невысокой плотностью населения на большей части страны и плохой транспортной доступностью значительной части населенных пунктов, приводят к тому, что значительную часть работ по оказанию первичной медицинской помощи «на местах» приходится выполнять медицинским волонтерам, которым постоянно требуется структурированная информационная поддержка их действий. Но, несмотря на высокое развитие телемедицинских технологий по стране в целом, подобная система информационной поддержки муниципальных медицинских волонтеров «на местах» до сих пор не создана. Поэтому, для создания такой системы, получившей название PersonalMedHelper (PMH), организована инициативная группа, в которую вошли студенты Университета ИТМО, СПбГУ и МГУ, обучающиеся по медицинским, телекоммуникационным и информационным направлениям, необходимым для решения этой задачи.

На самых первых шагах создания концепции телемедицинских продуктов PMH была поставлена задача построения корректного механизма определения экономического эффекта, реализующегося применительно к медицинско-гуманитарной сфере, оперирующей по большей частью неэкономическими показателями [1, 4].

Проведенный предварительный анализ возможных путей учета мультипликативного эффекта, достигающегося повышением доступности медицинских услуг, показал, что наилучшим образом «неэкономические» качества эффективности применения телемедицинских технологий могут быть учтены (в том числе и с учетом будущих периодов) с использованием концепции устойчивого развития [6]. Парадигмой этой концепции является утверждение возможности лишь такого развития, при котором удовлетворение потребностей нынешних поколений осуществляется без ущерба для возможностей будущих поколений удовлетворять свои собственные потребности.

Суть устойчивого развития в том, чтобы развитие нынешнего поколения не шло вразрез с интересами будущих поколений. В классическом определении баланс между поколениями понимается как баланс потребностей. Однако слово «потребности» может трактоваться по-разному.

Во-первых, можно говорить о способности удовлетворять потребности в контексте обеспеченности ресурсами. Это значит, что мы должны оставить будущим поколениям такое же количество ресурсов, что есть у нас, чтобы они с помощью этих ресурсов могли удовлетворять свои потребности.

Во-вторых, можно говорить о способности удовлетворять потребности как об уровне жизни. Это значит, что для будущих поколений он должен быть как минимум таким же, как уровень жизни нашего поколения. При этом совершенно необязательно, чтобы это было достигнуто благодаря обеспеченности ресурсами. Например, мы можем сейчас инвестировать ресурсы в развитие человеческого капитала и знаний, и будущие поколения будут обеспечивать более высокий уровень жизни за счет прироста знаний. Две перечисленные трактовки положили начало двум ветвям устойчивого развития – концепциям сильной и слабой устойчивости [7].

Укрупненно цели в области устойчивого развития [8] включают набор из более чем 170 задач по разным направлениям, объединенных в 17 целей – от преодоления бедности до обеспечения гендерного равенства, и принятия срочных мер по борьбе с изменением климата. Цели устойчивого развития сформулированы широко, и каждая из них конкретизирована в большом наборе различных показателей. При этом документ не предусматривает, каким образом они будут достигнуты. У международных институтов нет ни средств, ни полномочий на разрешение указанных проблем. Значительная часть финансирования может осуществляться только на национальном уровне. Цели устойчивого развития – это констатация указанных проблем и их приоритетности.

В целом, давая гносеологическую основу новому подходу к определению эффективности проводимых сегодня мероприятий, понятие «устойчивое развитие» вводит в рассмотрение и абстрактную идею, и точный показатель, один взгляд на который позволил бы ответить на вопрос, устойчиво ли развивается страна или регион в целом.

Важно понимать, что цели в области устойчивого развития не относятся к разряду плановых показателей, существование которых оправдано только их выполнением. Стоит, скорее, воспринимать их как ориентиры, к которым стоит стремиться [8]. Таким образом, оперировать терминами, относящимся к категории устойчивого развития особенно удобно в тех случаях, когда ведется стратегическая проработка концептуальных решений, требующих для реализации значительных финансовых и временных затрат.

Таким образом, в терминах устойчивого развития можно рассматривать эффективность внедрения телемедицинских систем (подобных создаваемому семейству телемедицинских продуктов РМН), поскольку, требуя больших вложений ресурсов уже сегодня, основной эффект использования таких систем окажется реализован лишь спустя значительное время. При этом совершенно необходимо, чтобы капиталовложения в развитие технологий помимо отложенного эффекта в будущих периодах, приносили экономический эффект и в экономически обозримом (т.е. подверженном текущему и среднесрочному планированию) будущем.

Исходя из учета этого обстоятельства создание продуктов семейства РМН началось с реализации проекта создания информационной системы поддержки муниципальных медицинских волонтеров, разработка которой не требует значительных затрат, но приводит к возникновению качественного эффекта повышения доступности парамедицинской помощи. В результате реализации проекта появится «персональный телемедицинский ассистент пациента», позволяющий на основе имеющихся передовых медицинских методик и достижений информационных технологий оказать своевременную квалифицированную предварительную информационную поддержку человеку в области охраны его здоровья (диагностика, лечение, уход), в тех случаях, когда он не может в силу тех или иных причин обратится в медицинское учреждение или хочет самостоятельно получить независимое мнение по медицинским вопросам, касающимся его или его близких.

Концепцию устойчивого развития, позволяющую учесть материальные и нематериальные выгоды будущих поколений, используют для учета эффективности капиталовложений, при расчетах наиболее амбициозных информационных проектов в области телемедицины [8]. Вкладывая сегодня значительные средства в сбор по всему миру однозначно избыточный (обработка которого в необходимом объеме сегодня технологически затруднена) объем фактографической медицинской информации, разработчики, исходя из основных положений концепции устойчивого развития, рассчитывают на мультипликативный отложенный эффект своих сегодняшних действий.

Однако следует обратить внимание на то, что оценкам эффективности в рамках концепции устойчивого развития порой не хватает конкретики [8]. Концепция устойчивого развития в своем текущем виде во многом является отражением сложности общества в целом – динамику его развития становится все сложнее перевести в сухие цифры. Причем цели в области устойчивого развития часто критикуют за нереалистичность и необеспеченность

сформулированных в них задач адекватными финансовыми ресурсами [9]. Важно понимать, что цели в области устойчивого развития не относятся к разряду плановых показателей, существование которых оправдано только их выполнением. Стоит скорее воспринимать их как ориентиры, к которым стоит стремиться [8].

Для построения более четкой модели устойчивого развития телемедицинского проекта (в том числе и РМН) необходимо учитывать методы, использующиеся в отраслях, в которых, согласно общей модели устойчивого развития [6] предполагается получение наибольшего эффекта от рассматриваемого проекта.

Для РМН согласно [8] предполагается достижение в трех (из 17) групп целей устойчивого развития, таких как:

- хорошее здоровье и благополучие (цель устойчивого развития № 3 по [8]);
- индустриализация, инновации и инфраструктура (цель № 9 по [8]);
- устойчивые города и населенные пункты (цель устойчивого развития № 11 по [8]).

В связи с тем, что на настоящий момент количественные методики расчета эффективности проектов, ориентированные на достижение целей устойчивого развития № 3 и № 4 находятся лишь на стадии разработки, то авторами предложено для расчета «отложенной» эффективности создания семейства продуктов РМН использовать методики, учитывающие особенности устойчивого развития городов и населенных пунктов. Поскольку обеспечение доступности высокотехнологичной медицинской помощи для населения является одним из важнейших факторов повышения качества жизни, однозначно связанным с экономической привлекательностью конкретного региона.

Таким образом, используя производные понятия концепции устойчивого развития, возможно найти подходы к построению оценочной модели «отложенного» экономического эффекта разработки и внедрения телемедицинских систем семейства в терминах показателей устойчивого развития территорий.

В результате проведенного анализа существующих методов оценки эффективности разработки и внедрения сложных телемедицинских систем, создание которых требует больших затрат и при использовании которых предполагается значительный отложенный эффект, установлено:

- помимо прямого экономического эффекта, выражающегося в сокращении издержек и повышении оперативности и качества медицинского обслуживания населения, при внедрении телемедицинских систем существует существенный «отложенный» мультипликативный эффект, связанный с повышением финансовой привлекательности регионов, обусловленной повышением качеством жизни населения;
- отложенный эффект от внедрения телемедицинских систем настолько значителен, что его необходимо учитывать при оценке эффективности проектов создания и внедрения телемедицинских систем;
- отложенный эффект может быть учтен при использовании концепции устойчивого развития городов и населенных пунктов, предполагающей комплексный учет текущих воздействий в течение длительного дальнейшего периода.

Причем следует отметить, что существующие сегодня подобного рода разработки методик расчета отложенного эффекта, базирующиеся на концепцию устойчивого развития, требуют внимательного изучения и обязательной верификации получаемых результатов практически для каждого случая их применения. Объясняется это тем, что предложенные решения не носят всеобъемлющего характера и представляют собой оценку лишь по ограниченному числу учтенных параметров.

Таким образом, при создании пациенто-ориентированных телемедицинских продуктов семейства РМН представляется целесообразным использовать механизмы концепции устойчивого развития для оценки отложенного эффекта от внедрения системы.

Корректный учет отложенного экономического эффекта от внедрения телемедицинских систем позволит существенно увеличить качество эффективности капиталовложений в разрабатываемые технологии, а следовательно, сделает их более привлекательными для возможных инверторов.

Литература

1. Андрющенко Г.И., Потехина Е.В. Экономические предпосылки развития рынка услуг здравоохранения в Российской Федерации // Социальная политика и социология. – 2015. – Т. 14. – № 5. – С. 6–14.
2. Сисигина Н. Развитие инфраструктуры здравоохранения в 2014–2016 гг. // Экономическое развитие России. – 2018. – № 5. – С. 68–72.
3. Шевлякова В.А. Управление процессом капитализации социального здоровья в условиях городской среды // Социальное здоровье: проблемы и решения. – 2018. – № 2. – С. 5–17.
4. Руденко М.Н., Гаухберг Л.Д. К вопросу об оценке эффективности системы здравоохранения // Национальные интересы: приоритеты и безопасность. – 2012. – № 17. – С. 57–64.
5. Перспективы телемедицины: выживут крупные и узкопрофильные IT-компании [Электронный ресурс]. – Режим доступа: https://telemedicina.ru/news/russian/perspektivy-telemediciny-vyzhivut-krupnye-i-uzkoprofilnye-it-kompanii (дата обращения: 01.11.2018).
6. Аткинсон Л. Как устойчивое развитие может изменить мир. – М.: Бином. Лаборатория знаний, 2012. – 455 с.
7. Ягодин Г.А. Устойчивое развитие. Человек и биосфера. – М.: Бином, 2013. – 109 с.
8. Крылов А.Н. Корпоративная социальная ответственность: экономические модели – мораль – успех – устойчивое развитие. – М.: ИКАР, 2013. – 452 с.
9. Марфенин Н.Н. Устойчивое развитие человечества. – М.: МГУ, 2007. – 624 с.

Мерзлякова Юлия Валерьевна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41601
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: mer.yulia@mail.ru

Готская Ирина Борисовна
Год рождения: 1957
Университет ИТМО, факультет программной инженерии и компьютерной техники, д.педагог.н., профессор
e-mail: iringot@mail.ru

УДК 37.026.4

ИССЛЕДОВАНИЕ ДИЗАЙНА ИНФОГРАФИКИ ДЛЯ ПОДДЕРЖКИ УРОКОВ В ОСНОВНОЙ ШКОЛЕ

Мерзлякова Ю.В.
Научный руководитель – д.педагог.н., профессор Готская И.Б.

В работе представлены результаты исследования проблемы дизайна инфографики. Определена роль инфографики для поддержки уроков в основной школе, а также некоторые принципы, которым должен соответствовать дизайн инфографических данных. Доказана не изученность вопроса о дизайне инфографики в образовании, в частности в основной школе.
Ключевые слова: инфографика, визуализация, дизайн, инфографические данные, информация, учебный процесс.

Для современного общества характерно возрастание объемов информации, которые ежедневно приходится усваивать каждому человеку. Рост информационной активности приводит к появлению новых способов представления информации. Инфографика – один из графических способов представления информации, с помощью которого данные визуализированы в виде образов. Таким образом, инфографика позволяет визуализировать информацию, что способствует повышению эффективности ее восприятия и усвоения.

В научных исследованиях [1–3] обосновано, что дизайн является важной составляющей при формировании визуального образа графического изображения, оказывает влияние на его восприятие субъектом и определяет уровень наглядности данных.

Говоря о дизайне инфографики, следует уточнить понятие «инфографика», однозначного толкования которого не существует. Однако на основе анализа научных исследований [1, 2, 4] можно сделать вывод, что инфографика – это средство визуальной односторонней коммуникации, представляющее сложную информацию, данные в сжатом, структурированном виде, что позволяет визуализировать большие объемы информации, облегчая при этом процесс ее восприятия.

Исходя из трактовки понятия «инфографика», ее можно рассматривать как одно из современных эффективных средств обучения, способствующих повышению наглядности, и одновременно как специфический вид визуальной коммуникации. Использование инфографики в учебном процессе способствует решению ряда важных педагогических задач:

- передача большего объема научной информации обучающимся в тот же промежуток времени, без снижения качества обучения и качества передаваемой информации;
- повышение интереса обучающихся к учебной познавательной деятельности;
- развитие восприятия, абстрактного, логического и визуального мышления;
- формирование навыков распознавания образов и получения новой учебной информации из представленных образов на основе их анализа.

Следует заметить, что характерной особенностью обучения в основной школе является развитие абстрактно-логического мышления обучающихся, что подтверждает педагогическую целесообразность использования инфографики в основной школе, особенно для изучения естественнонаучных учебных дисциплин (физика, химия, биология) и технологии [2].

Однако все перечисленные достоинства инфографики будут эффективны в обучении, только при разработке дизайна инфографических данных, соответствующего определенным научно-обоснованным требованиям.

Дизайн – это процесс проектирования, создания, творческая деятельность, направленная на получение результата [5]. Дизайн инфографики должен визуализировать данные таким образом, чтобы привлекать внимание субъекта к наиболее важным деталям, при этом данные должны отображаться правильно, без нарушений логической структуры визуальными эффектами. Привлекательный и понятный дизайн инфографики будет способствовать восприятию представленной информации, ее освоению и лучшему пониманию с целью получения новых знаний, а не препятствовать этому процессу.

В научных работах ряда авторов [1–3], посвященных исследованию проблемы дизайна инфографики, определены базовые подходы создания дизайна, не зависящие от сферы применения инфографики, а также сформулированы некоторые принципы, которым должен соответствовать дизайн инфографики, независимо от сферы ее применения. Рассмотренные подходы и принципы могут быть адаптированы для разработки дизайна учебной инфографики.

- Излишняя визуальная выразительность или напротив, отсутствие привлекательности и гармонии, окажут негативное влияние на процесс изучения информации с помощью инфографических данных. Упор на визуальные качества и отсутствие структуры, ясности, простоты восприятия лишает смысла внедрения инфографики в учебный процесс.
- Учебная инфографика должна не только графически заинтересовывать, но и включать соответствующую структуру, точное содержание, обладать глубиной научных знаний [4].
- Визуальные составляющие инфографики должны поддерживать предоставленную информацию, создавая ассоциативную связь между графическими элементами дизайна и их смысловым содержанием.
- Дизайн должен способствовать усвоению нового учебного материала, а не замедлять его.

Хороший дизайн инфографики поэтапно и структурированно представляет информацию благодаря сочетанию дизайна визуализации данных и графическому дизайну. Дизайнеры инфографики используют визуализацию данных и иллюстрации как визуальный компонент. Яркая, привлекающая внимание визуализация заставляет аудиторию запоминать представленную информацию. При этом первостепенным должна оставаться структура, точность, функциональность, ясность.

О проблеме равновесии между эстетической привлекательностью и функциональностью упоминает Э.Р. Тафти, повествуя о том, что в современном мире существуют примеры, где способ визуализации информации заинтересовывает и обращает на себя гораздо больше внимания, чем сами знания, находящиеся под визуальной оболочкой [3]. Такая ошибка в дизайне инфографики для основной школы повлечет снижение эффективности учебного процесса.

Анализируя важность визуализации данных на процесс познания, обучения, можно обратиться к истории. Хорошим примером являются важнейшие открытия Галилея в области астрономии, которые ему удалось совершить благодаря правильной визуализации своих наблюдений и данных. Выводы и открытия были сделаны об астрономических объектах, основывались на зарисовках ученого в течение некоторого промежутка времени. Неточность же изображений Галилея, выполненных при изучении проблем с солнечными пятнами, заставила усомниться научное сообщество в правильности его теории [4]. Данный пример еще раз доказывает насколько визуализация, точность и ясность передачи информации важны при ее использовании в качестве источника научных знаний.

Однако вопрос дизайна в образовании, в частности в основной школе, на данный момент остается не до конца изученным, некоторые авторы в своих научных исследованиях [1, 4] говорят о том, что требования к дизайну инфографики в учебном процессе обозначены поверхностно. Нет четко сформулированных критериев о том, каким должен быть дизайн инфографики, чтобы повысить эффективность обучения, сделать сложную для восприятия информацию более понятной и структурированной.

Х.М. Фадзил упоминает в своем исследовании о не изученности визуальных вопросов особенной обучающей инфографики [4].

Из исследования Т.Ю. Быстровой можно сделать вывод, что отсутствие четких требований к дизайну образовательной инфографики приводит к тому, что часто визуализированные данные не просто не повышают эффективность, но и даже наносят вред усвоению новой информации [1].

Проведенное исследование позволяет сделать вывод о том, что проблема создания дизайна инфографики для целей основного образования остается не до конца изученной, требования к дизайну инфографики в учебном процессе сформулированы недостаточно четко. Выделены критерии для разработки дизайна инфографики, применимые для создания инфографики для различных сфер жизнедеятельности человека. Однако эти критерии не учитывают специфики использования инфографики в основном образовании, психофизиологических особенностей восприятия учащихся и т.д., что подтверждает актуальность проведения дополнительных исследований.

Литература

1. Быстрова Т.Ю. Дизайн эффективной инфографики для МООК: определение значимых характеристик (Часть 1) // Академический вестник УралНИИПроект РААСН. – 2017. – № 3(34). – С. 92–98.
2. Суханова Д.В. Критерии оценки дизайна как продукта дизайна // Вестник современных исследований. – 2018. – № 5.1(20). – С. 200–206.
3. Tufte E.R. Envisioning Information. – Cheshire, CT: Graphics Press, 1990. – 126 p.
4. Fadzil H.M. Designing infographics for the educational technology course: perspectives of preservice science teachers // Journal of Baltic scince education. – 2018. – V. 17. – № 1. – P. 8–18.
5. Naparin H., Saad A.B. Infographics in education: rewiew on infographics design // The international journal of multimedia and its applications. – 2017. – V. 9. – № 4/5/6 (Desember). – P. 15–24.

Меркулова Ольга Владимировна
Год рождения: 1994
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4260
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: olga.merkulova094@gmail.com

Готская Ирина Борисовна
Год рождения: 1957
Университет ИТМО, факультет программной инженерии
и компьютерной техники, д.педагог.н., профессор
e-mail: iringot@mail.ru

УДК 004.9

СРАВНИТЕЛЬНЫЙ АНАЛИЗ JAVASCRIPT-ФРЕЙМВОРКОВ

Меркулова О.В.
Научный руководитель – д.педагог.н., профессор Готская И.Б.

В работе приведен сравнительный анализ трех самых популярных по состоянию на 2019 год JavaScript-фреймворков, выделенных путем анализа запросов Google и количества размещенных вакансий на различных ресурсах. Рассмотрены ключевые особенности, достоинства и недостатки, а также их сравнение по различным критериям, а также выделены рекомендации по использованию их в различных проектах.
Ключевые слова: JavaScript, веб-разработка, создание сайтов, JavaScript-фреймворки, сравнение JavaScript-фреймворков.

В настоящее время наблюдается стремительное развитие веб-разработок: в онлайн уходит обучение, общение, бизнес, медицинское консультирование, государственные и финансовые услуги и т.д. Ввиду этого разрабатывается большое количество веб-сайтов, при создании которых разработчики часто используют различные фреймворки и библиотеки для упрощения процесса разработки.

Фреймворки – это программные продукты, которые упрощают создание и поддержку технически сложных или нагруженных проектов. Как правило, фреймворк содержит только базовые программные модули, а все специфичные для проекта компоненты реализуются разработчиком на их основе. Тем самым достигается не только достаточно высокая скорость разработки, но и большая производительность и надежность решений [1]. Как следствие, фреймворки стремительно развиваются и активно используются веб-разработчиками.

Рассмотрим три JavaScript-фреймворка: React.js, Vue.js и Angular.js. Анализируя их популярность по количеству запросов Google по всему миру за последние 5 лет, очевидно, что долгое время самым популярным среди рассматриваемых фреймворков по запросам был Angular, и в последний год его догоняет и обгоняет React.js [2]. При этом Vue.js от них отстает и набирает популярность достаточно медленно (рис. 1).

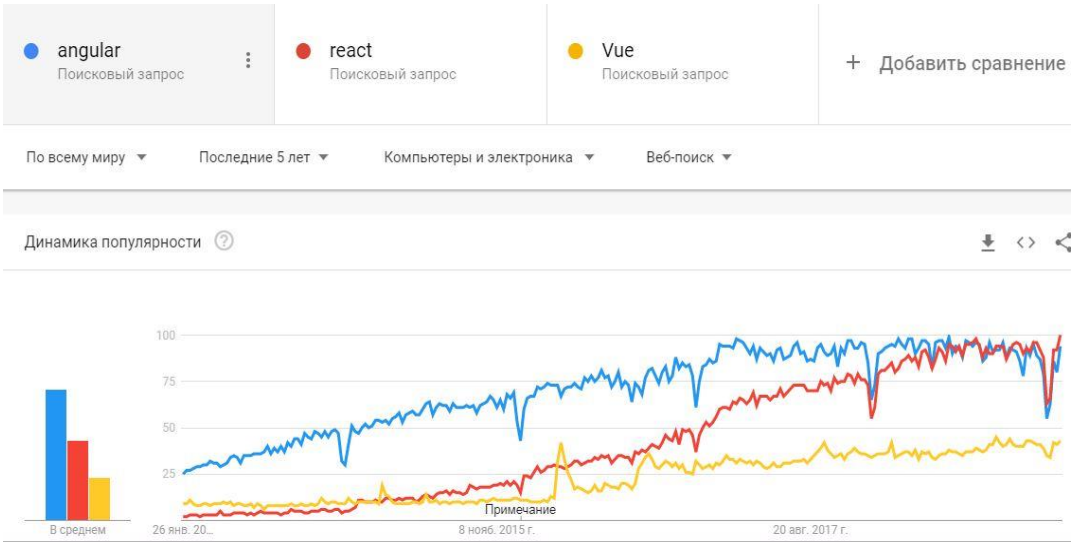


Рис. 1. Популярность JavaScript-фреймворков в мире

Ниже представлено количество опубликованных вакансий по данным на конец декабря 2018 года на различных ресурсах [2]. Из графика видно, что ситуация примерно соответствует поисковым запросам (рис. 2).

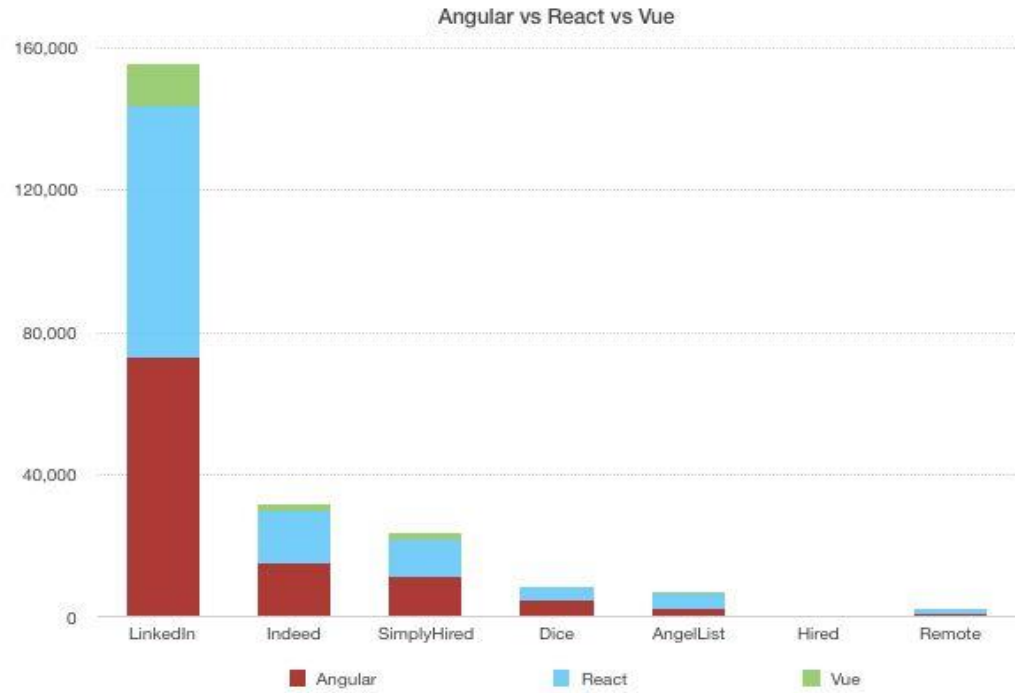


Рис. 2. Опубликованные вакансии, требующие знание фреймворков в мире

React – эффективная и гибкая библиотека JavaScript для генерации динамических пользовательских интерфейсов. Его используют в Uber, Instagram, Twitter, Wix, PayPal и др. React подходит для проектов, где важна гибкость при создании больших систем, имеющих тенденцию к разрастанию, а также для командной разработки. Преимуществом React является читабельный и простой в сопровождении код [3].

Однако использование библиотеки React требует базовых знаний языка JavaScript. Данный фреймворк позволяет повторно использовать код, что позволяет не только ускорить разработку, но и избежать большого количества ошибок. React развивается как открытый проект, и все сайты библиотеки доступны в интернете. Кроме того, существует сайт с официальной документацией на английском языке, где можно найти все необходимые руководства по использованию [3].

React лучше всего подходит для создания крупных веб-приложений, использующих данные и их изменения со временем без перезагрузки страницы. Приложения на React можно разработать достаточно быстро, и они более масштабируемы. React использует абстрактную копию реального DOM и обновляет изменения, сделанные пользователем, без затрагивания других частей веб-страницы. Однако данный фреймворк не поддерживается браузерами Internet Explorer ниже 8 версии [4].

Одним из недостатков React.js является открытый исходный код и ежедневное появление новых библиотек, что осложняет поиск документации конкретной свежей выпущенной библиотеки. Однако есть возможность подключать к приложению различные версии фреймворка [5].

Angular – это платформа с открытым исходным кодом может быть использована для разработки одностраничных веб-приложений и мобильных приложений. Angular помогает справляться с многочисленными запросами данных. Однако фреймворк достаточно сложный в силу своих многовариантных решений, и его изучение занимает достаточно много времени [4]. Следует выделить одно из преимуществ этого фреймворка в сравнении с React – очень подробная документация.

Прямой переход с одной версии Angular на другую невозможен, т.е. необходимо последовательно устанавливать все обновления между версиями [5].

Angular спроектирован как универсальный фреймворк, поэтому на нем можно создавать приложения практически любого типа. Кроме того, приложения, написанные на Angular, как правило, компактные и легкие в редактировании [6].

В то же время Angular избыточен для некоторых проектов, недостаточно гибкий (например, в сравнении с React и Vue), а также этот фреймворк нельзя использовать для обработки манипуляций с большим количеством данных, поскольку любое изменение влечет за собой изменение DOM, т.е. редакторы и видеоигры будут работать очень плохо.

Vue – это JavaScript-фреймворк с открытым исходным кодом, используемый для создания пользовательских интерфейсов. Он полезен при создании одностраничных приложений и внесения компонентов в существующие приложения [4].

К недостаткам Vue.js относятся малое количество документации на русском и английском языках, а также небольшое сообщество пользователей [5].

Результаты сравнительного анализа фреймворков представлены в таблице.

Таблица. Результаты сравнительного анализа JavaScript-фреймворков

Критерий	React.js	Vue.js	Angular
Скорость работы	Быстрая, Vue.js чуть быстрее из-за более легкого веса		Медленнее
Легкость изучения	Достаточно легко	Легко	Достаточно сложно
Размер исходного файла	626 КБ	303 КБ	1,3 МБ
Наличие документации	Очень быстрое обновление, сложно найти документацию	Малое количество документации на русском и английском языках, скромное сообщество пользователей	Очень подробная, большое количество разработчиков
Совместимость версий	Можно подключать любую версию фреймворка		Не совместимы

Результаты проведенного анализа показали, что для приложений, требующих большого количества изменений, подходит React или Vue, которые, к тому же занимают меньше места и работают быстрее, чем Angular. Vue, в сравнении с React, имеет меньший вес и более легкий порог вхождения при схожих принципах работы, но ввиду большой популярности React

решения различных вопросов легче найти именно в отношении этого фреймворка. Однако для приложений, не требующих множества изменений DOM и последующей модификации проекта, Angular очень хорошо подходит из-за дружественного интерфейса, подробной документации и большого сообщества разработчиков.

Литература

1. Фреймворки в веб-разработке, 2014 [Электронный ресурс]. – Режим доступа: https://web-creator.ru/articles/about_frameworks (дата обращения: 26.01.2019).
2. React, Angular и Vue: что изучать в 2019 году? [Электронный ресурс]. – Режим доступа: https://tproger.ru/translations/react-angular-vue-for-2019/ (дата обращения: 26.01.2019).
3. Бондаренко С.О. Современные интерактивные веб-приложения – построение пользовательского интерфейса с React // Вестник науки и образования. – 2018. – № 5(41). – Т. 2. – С. 46–48.
4. 6 фреймворков веб-разработки, которые будут в тренде в 2018 [Электронный ресурс]. – Режим доступа: https://techrocks.ru/2018/01/30/trend-web-development-frameworks-for-2018/ (дата обращения: 25.01.2019).
5. Сравнение фреймворков Vue, React и Angular (2019) [Электронный ресурс]. – Режим доступа: https://mkdev.me/posts/sravnenie-freymvorkov-vue-react-i-angular-2019 (дата обращения: 25.01.2019).
6. 18 советов по повышению производительности: введение в Angular JS [Электронный ресурс]. – Режим доступа: https://webformyself.com/18-sovetov-po-povysheniyu-proizvoditelnosti-vvedenie-v-angular-js/ (дата обращения: 26.01.2019).

Миннемуллин Булат Мансурович
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4262
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: b@minbulat.ru

Готская Ирина Борисовна
Год рождения: 1957
Университет ИТМО, факультет программной инженерии и компьютерной техники, д.педагог.н., профессор
e-mail: iringot@mail.ru

УДК 004.588

ИГРОФИЦИРОВАННОЕ ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ДИСТАНЦИОННОГО ОБУЧЕНИЯ ШКОЛЬНИКОВ

Миннемуллин Б.М.
Научный руководитель – к.педагог.н., профессор Готская И.Б.

В работе представлены способы взаимодействия пользователя с приложением, разработана архитектура информационной системы, описываются проект базы данных и подсистемы приложения.
Ключевые слова: игрофикация, информационная система, архитектура приложения, веб-приложение, чат-боты.

Современные приложения предоставляют пользователю разные способы взаимодействия: через браузер, настольные приложения для разных операционных систем, мобильные приложения. Также в последнее время получили распространение чат-боты для разных мессенджеров, позволяющие организовать быстрое взаимодействие с информационной системой [1]. Обучение школьников с использованием игрофикации будет эффективнее, если у ученика будет возможность обучаться в разных местах и с разных устройств. Реализация такой системы требует хорошей организации программного кода приложения: разделение программы на модули, использования различных библиотек и фреймворков, проверку взаимной работоспособности различных частей системы.

Архитектура программного обеспечения – это совокупность решений для организации программной системы, включающая выбор структурных элементов, их интерфейсов и их поведения в рамках взаимодействия структурных элементов; соединение выбранных элементов структуры и поведения в более крупные системы; а также архитектурный стиль, создающий основу для всех элементов системы и их интерфейсов, их взаимодействие и соединение [2, 3].

Разрабатываемая информационная система, организовывающая самостоятельное дистанционное обучение школьников, используя игрофикацию, состоит из разных пользовательских интерфейсов (веб-приложение, настольные приложения для разных операционных систем, мобильные приложения, чат-боты и голосовой помощник). Каждый пользовательский интерфейс рассматривается как отдельное приложение. У всех этих приложений есть общие части (база данных (БД), бизнес-логика) и различающиеся элементы

(пользовательские интерфейсы, особенности взаимодействия). Поэтому информационная система состоит из нескольких подсистем, представленных на рисунке.

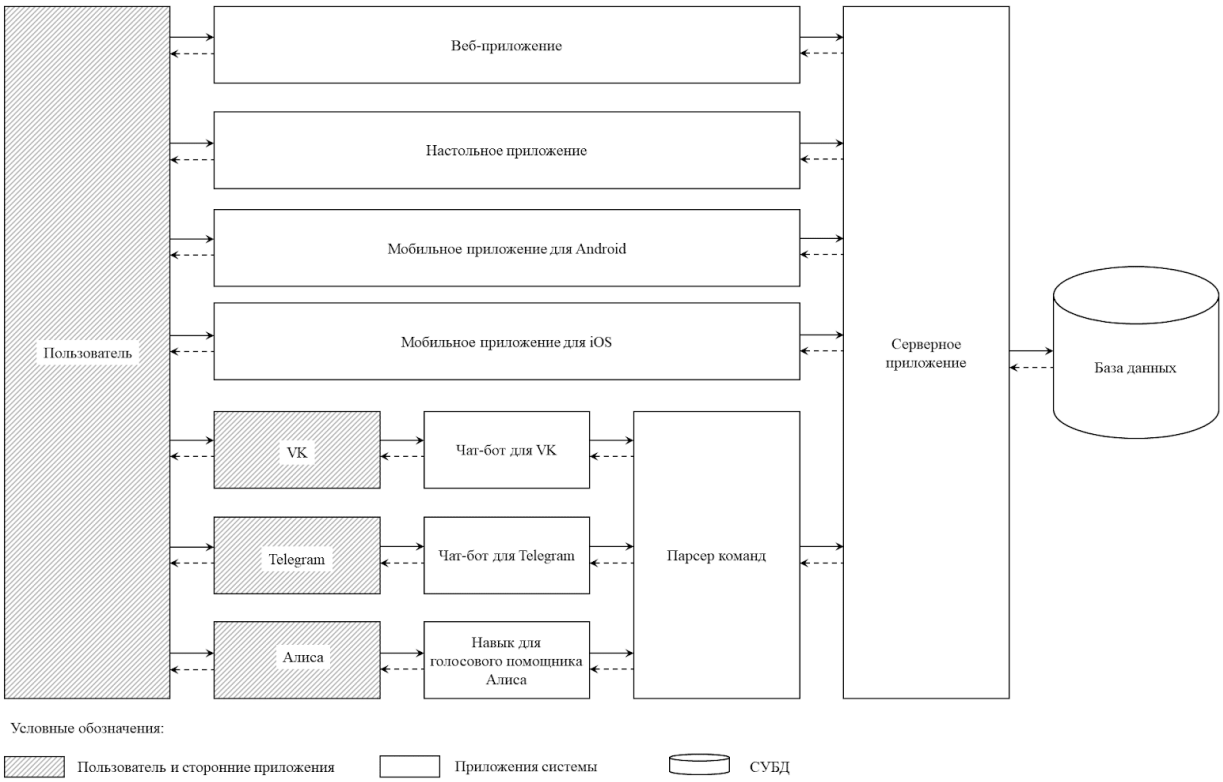


Рисунок. Архитектура приложения

БД хранит данные приложения. Используется графовая система управления базами данных (СУБД) с открытым исходным кодом Neo4j. В ней элементы содержатся в виде графа, который состоит из узлов (nodes) и связей (relations). Узлы и связи могут содержать свойства (properties), которые можно использовать для хранения информации. В СУБД используется собственный язык запросов – Cypher.

Серверное приложение осуществляет получение данных из БД, обработку и выдачу клиентским приложениям, а также обратную операцию – получение данных от клиентских приложений, обработку и внесение их в БД. В серверном приложении реализуется бизнес-логика системы. Оно пишется на языке JavaScript (Node.js) и основана на фреймворке Express.

Клиентское веб-приложение загружается в пользовательское устройство через браузер при открытии веб-страницы и отображает графический интерфейс пользователя. Пишется на языке JavaScript и основана на фреймворке Vue.js.

Клиентское настольное приложение запускается в операционных системах Windows, Linux и macOS и отображает графический интерфейс пользователя. Пишется на языке JavaScript, создается на основе веб-приложения, основана на фреймворке Electron.

Мобильные приложения создаются отдельно для каждой мобильной операционной системы (Android и iOS), загружаются из магазина приложений (Play Market для Android и AppStore для iOS) и отображают графический интерфейс пользователя. Создаются на основе веб-приложения с помощью библиотеки NativeScript, которая позволяет писать мобильные приложения на JavaScript, используя фреймворк Vue, который в информационной системе используется при разработке web-клиента приложения. Благодаря этому процесс разработки упрощается, появляется возможность переиспользования исходного кода web-приложения для мобильных приложений. NativeScript генерирует приложения для Andriod и iOS.

Чат-боты позволяют взаимодействовать с пользователем через мессенджеры. Пользователь отправляет текстовые команды через мессенджеры и получает ответы в виде

текста, картинки или видео. Реализуются чат-боты для мессенджера социальной сети ВКонтакте и мессенджера Телеграмм.

Голосовой интерфейс позволяет пользователю отдавать команды голосом. Реализуется интерфейс на платформе Яндекс.Коннект (навык для голосового помощника Алиса).

Мессенджеры предоставляют похожий функционал для взаимодействия пользователем, поэтому чат-боты и голосовой интерфейс осуществляют взаимодействие с серверным приложением не напрямую, а через промежуточное приложение-парсер запросов. Это приложение обрабатывает текстовые команды, поступающие от чат-ботов и голосового интерфейса, и отправляет их серверному приложению.

Взаимодействие приложений между собой осуществляется через API. Серверное приложение и СУБД взаимодействуют по протоколу bold, остальные приложения взаимодействуют между собой через REST API. Архитектурный стиль REST предоставляет возможность создания кроссплатформенных, масштабируемых систем, гибких к изменению структуры и повышению производительности [4]. REST API устанавливает характер взаимодействия с WEB API посредством базовых операций (CRUD: create, read, update, delete) с использованием HTTP-протокола [5].

Все приложения обмениваются между собой документами JSON. Формат передаваемых данных между сторонними приложениями описан в их документациях, а между разрабатываемыми приложениями составляется самостоятельно.

Информационная система для игрофицированного дистанционного обучения школьников состоит из нескольких подсистем:

1. подсистема обучения;
2. подсистема проверки знаний;
3. подсистема интерфейса;
4. подсистема игрофикации.

Подсистема обучения организует демонстрацию различных обучающих материалов обучающимся: текстов, видео, слайдов и других видов обучающих материалов.

Подсистема проверки знаний реализует проведение тестирования обучающихся. Данная подсистема формирует проверочные задания, выводит их пользователю и осуществляет проверку правильности введенных ответов.

Подсистема игрофикации организует начисление баллов за выполнение заданий, составление таблиц лидеров, проведение соревнований между пользователями, формирование ежедневных и еженедельных заданий и другие операции, связанные с реализацией игрофикации.

Подсистема интерфейса позволяет разным частям информационной системы взаимодействовать между собой, а также предоставляет пользователю возможность регистрации и авторизации, реализует различные пользовательские интерфейсы.

При развертывании приложения используется веб-сервер Nginx, запущенный в контейнерах Docker.

Nginx используется в качестве прокси и фронтенда на сервере. Программа получает пользовательские запросы и распределяет их среди подсистем приложения: возвращает статические файлы (файловый сервер) или запускает приложение на Node.js (бекэнд-сервер).

Let’s Encrypt позволяет получать бесплатные сертификаты для создания SSL-соединения. Использование данных сертификатов позволяет подключаться к информационной системе протоколом HTTPS. Протокол HTTPS обеспечивает шифрование передаваемых данных, что позволяет уменьшить вероятность получения сторонними лицами конфиденциальной информации и подмены содержимого передаваемой информации.

Dockers используется для создания контейнеров, которые запускаются на сервере. Данная технология позволяет формировать образы, которые будут одинаково работать независимо от компьютера, на котором запускаются.

Docker compose позволяет объединить несколько docker-контейнеров для их совместной работы.

Современные приложения предоставляют пользователю разные способы взаимодействия (через браузер, настольные и мобильные приложения, мессенджеры, голосовые помощники). Разработка таких приложений требует грамотного проектирования их архитектуры. Разработанные проекты БД, серверного приложения для получения информации из БД и организации бизнес-логики информационной системы создадут основу для разработки веб-интерфейса, настольного и мобильных приложений.

Литература

1. Смыслова Л.В. Чат-бот как современное средство интернет-коммуникаций // Молодой ученый. – 2018. – № 9. – С. 36–39.
2. Kruchten P. The Rational Unified Process: An Introduction. – 2nd Ed. – Addison-Wesley, 2000. – 320 p.
3. Rumbaugh J., Jacobsen I., Booch G. The Unified Modelling Language Reference Manual. – 2nd Ed. – Boston: Addison-Wesley, 2004. – 742 p.
4. Richardson L., Amundsen M. RESTful Web APIs. – Sebastopol: O’Reilly Media, 2013. – 94 p.
5. Егунов В., Князев А. Проектирование оптимизированной архитектуры web-клиента, взаимодействующего с REST API // Изв. Волгоградского государственного технического университета. – 2018. – № 8. – С. 79–83.

Миронова Маргарита Михайловна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41701
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: mirronovar@gmail.com

Меженин Александр Владимирович
Год рождения: 1959
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: mejenin@mail.ru

УДК 000.00

ВИДЫ ПРЕЗЕНТАЦИОННОГО ВИДЕО В ТЕАТРЕ

Миронова М.М., Меженин А.В.
Научный руководитель – к.т.н., доцент Меженин А.В.

Работа выполнена в рамках темы НИР «Современные технологии создания презентационного видео в театре».

В работе рассмотрена актуальность использования видеозаписи в театре в рамках рекламного материала. Представлены варианты видеопродукции, которые возможны в реализации в театре, а также рассмотрены параметры и характеристики, которыми должен обладать видеоролик для восприятия аудиторией и достижения рекламных целей.
Ключевые слова: мультимедийные технологии, видео, VR, видео 360, трейлер.

Видео давно проникло в театральную сферу, только теперь оно работает не только на сцене, но и за ее пределами. Однако гораздо интереснее и важнее рассмотреть те актуальные проблемы, которые возникают в театре сегодня.

Данная тема актуальна по причине колоссальной скорости развития технологий, и театральная сфера не может игнорировать тот факт, что для успешного роста необходимо следовать трендам и изменениям, которые происходят в производстве видеоконтента.

Презентационное видео представляет собой видеопродукцию, которая находится в Интернете, в самом театре и позволяет зрителю ознакомиться со спектаклем, проявить интерес к нему и сложить некое мнение. Зрителя надо побудить к покупке, показать, что именно этот спектакль он должен посмотреть в ближайшее время, а лучше всего это сделать наглядно – т.е. показать трейлер к спектаклю.

Трейлер или тизер, как многие могут назвать, является мощным инструментом в побуждении интереса. Однако стоит оговориться, что между этими двумя понятиями есть разница. Хорошее объяснения понятия трейлер имеется в статье [1], где сказано, что «хорошие документальные трейлеры – это интересные отдельные лакомые кусочки развлечений, которые убеждают зрителей в том, что они хотят посмотреть полнометражный фильм, который «дразнит». Данное понятие можно применить и к театральной сфере, где фрагменты спектакля побуждают интерес у зрителя к данной постановке. В свою очередь, тизер, это некая приманка, которая побуждает зрителя задать себе вопрос «Что это?». Создать неизвестность,

непонимание, загадку – это основная задача тизера. В отличие от трейлера он снимается специально и не иллюстрирует сюжет спектакля, а скорее создает атмосферу. После просмотра тизера человеку необходимо будет понять, что же скрывается за этим коротким и, на первый взгляд, бессмысленным видеорядом.

В ходе работы были прослушаны лекции по производству видео, статьи на тему восприятия людьми видео с точки зрения психологии человека [2], а также изучены статистические данные [3], которые проводились по подобной проблеме специалистами за рубежом. Критерии, про которые уже можно сказать сейчас – это хронометраж, качество видеозаписи и звук. Это три константы, которые являются одними из самых важных составляющих качественного продукта. Так, человек воспринимает максимальное количество информации из видеоролика хронометражем до двух минут. Далее объем информации уменьшается. Относиться к звуку стоит очень внимательно, поскольку необходимо понимать роль аудиосопровождения. Это может быть информативная цель, атмосферное сопровождение или прямое дополнение к изображению.

Чтобы изучить направления, в которых можно создавать видеопродукцию, стоит обратиться к смежным темам, таким как телевидение, интернет, клип мейкинг, игры. Возможно, какая-то механика полностью подойдет для театральной сферы, какую-то необходимо будет подстроить под новые рамки, а что-то изменить вовсе. В работе рассмотрены видео 360 и VR (рисунок, а) [2, 4–11].

а б

Рисунок. Видео 360: панорамная прогулка по Большому театру (а); сцена из VR-спектакля «Клетка с попугаями» (б)

На данном изображении можно увидеть кадр тура по Большому театру, который находится на сайте театра и доступен всем желающим. У посетителей есть возможность пройтись по театру перед его посещением, посмотреть зал и выбрать для себя место.

Данное изображение (рисунок, б) – кадр из спектакля Максима Диденко, который создан с использованием очков виртуальной реальности. Зритель надевает очки и попадает в спектакль, где сам является частью представления.

В дальнейшем планируется рассмотреть другие примеры видеопродукции и проанализировать их эффективность путем эксперимента, для проведения которого будут взяты зрители театра.

Литература

1. The Importance of Documentary Trailers [Электронный ресурс]. – Режим доступа: https://www.thoughtco.com/the-importance-of-documentary-trailers-1109851 (дата обращения: 04.01.2019).
2. 85 percent of Facebook video is watched without sound [Электронный ресурс]. – Режим доступа: https://digiday.com/media/silent-world-facebook-video/ (дата обращения: 04.01.2019).

3. 40+ VIDEO MARKETING STATISTICS 2018 [INFOGRAPHIC] [Электронный ресурс]. – Режим доступа: https://www.renderforest.com/blog/video-marketing-statistics (дата обращения: 04.01.2019).
4. Видеомаркетинг в соцсетях: самые важные показатели в 2017 [Электронный ресурс]. – Режим доступа: https://habr.com/company/alconost/blog/326944/ (дата обращения: 04.01.2019).
5. Видео в digital-коммуникациях [Электронный ресурс]. – Режим доступа: https://www.cossa.ru/trends/201228/ (дата обращения: 04.01.2019).
6. Гайд по видео в контент-маркетинге. Часть 1: Концепция и создание [Электронный ресурс]. – Режим доступа: https://netology.ru/blog/videoguide (дата обращения: 04.01.2019).
7. Ильина Е.Л., Латкин А.Н., Бочарова Э.А. Видеомаркетинг как перспективное направление контент-маркетинга [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/n/videomarketing-kak-perspektivnoe-napravleniekontent-marketinga (дата обращения: 04.01.2019).
8. Канащук Т.Н. Маркетинг в сфере театрального искусства [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/n/marketing-v-sfere-teatralnogo-iskusstva (дата обращения: 04.01.2019).
9. Читать или смотреть? Видео в тренде: YouTube, Instagram и Facebook [Электронный ресурс]. – Режим доступа: https://www.cossa.ru/cmyk/145448/ (дата обращения: 04.01.2019).
10. Cisco Visual Networking Index: Forecast and Methodology, 2016–2020 [Электронный ресурс]. – Режим доступа: https://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/complete-white-paper-c11-481360.html (дата обращения: 04.01.2019).
11. How Long Should Your Next Video Be? [Электронный ресурс]. – Режим доступа: https://wistia.com/blog/optimal-video-length (дата обращения: 04.01.2019).

Молодкина Анастасия Александровна
Год рождения: 1993
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4272
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: em_6@mail.ru

Локалов Владимир Анатольевич
Год рождения: 1959
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: localov@gmail.com

УДК 004.51

ИНДИВИДУАЛЬНЫЕ ОСОБЕННОСТИ ВОСПРИЯТИЯ ВИЗУАЛЬНЫХ НАПРЯЖЕНИЙ КОМПЬЮТЕРНОГО КАДРА

Молодкина А.А., Локалов В.А.
Научный руководитель – к.педагог.н., доцент Локалов В.А.

Удобство расположения и качество передачи информации зависят от восприятия визуальных напряжений компьютерного кадра. Арнхейм полагал, что восприятие таких напряжений у всех людей одинаково. Для проверки этого положения был проведен эксперимент и оценена стабильность результатов. Эксперимент показал разные типы восприятия визуальных напряжений компьютерного кадра, которые обусловлены индивидуальным визуальным опытом и навыком визуального восприятия.
Ключевые слова: визуальное напряжение, компьютерный кадр, интерфейс, проектирование интерфейса, Арнхейм.

Одной из важнейших задач проектирования интерфейса человеко-машинного взаимодействия является нахождение визуально-сбалансированного расположения элементов интерфейса внутри экранного кадра.

Для решения этой задачи часто используется идея Арнхейма об априорно неравнозначной визуальной напряженности, которая меняется в зависимости от позиции объекта в кадре [1]. Визуальная напряженность объекта также воздействует на другие объекты, придавая зрительное направление их потенциальному движению и предлагая, куда смотреть дальше [2]. Визуализация поля с такими напряжениями может дать дизайнеру информацию о том, насколько объект будет казаться устойчивым или неустойчивым при размещении его в той или иной точке кадра.

Заметив различия в восприятии одной и той же визуальной структуры, было принято решение проверить степень универсальности идеи Арнхейма. Для этого в Университете ИТМО был проведен ряд экспериментов. Они выявили наличие существенных индивидуальных особенностей восприятия визуальных напряжений экранного кадра, которые существующие подходы не учитывают при разработке визуально сбалансированной структуры интерфейса [3, 4].

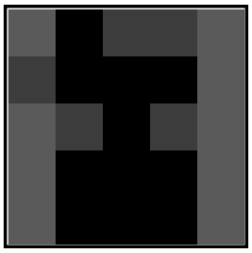
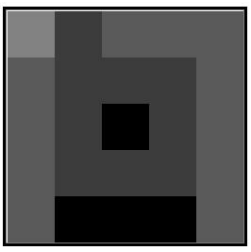
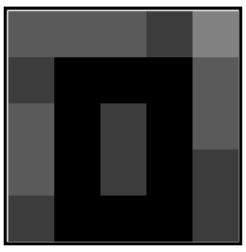
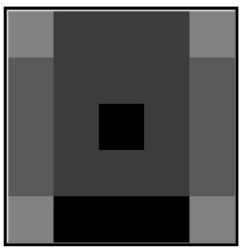
Для автоматизации эксперимента был разработан комплекс программ, который обеспечил возможность проведения исследований на большой группе респондентов, среди

которых в основном были студенты Университета ИТМО. На рис. 1 представлен пример теста на выявление индивидуальных особенностей восприятия визуальной напряженности – фигура располагается в одном из «секторов» поля. Испытуемому предлагалось оценить впечатление от изображения. После чего фигура располагалась в другом «секторе» поля и снова необходимо было дать оценку изображению.

Рис. 1. Пример теста

Результатом тестирования является поле с секторами, в котором серыми оттенками показаны напряженные, а черным – спокойные положения фигуры (таблица).

Таблица. Результаты двух тестирований одного респондента по двум фигурам

	Квадрат		Круг	
	Первое тестирование	Второе тестирование	Первое тестирование	Второе тестирование
Результаты респондента				

Далее каждый результат (поле) переводился в матрицу чисел от 1 до 4, где:
– черный сектор = 1 (спокойное положение фигуры);
– темно-серый сектор = 2 (почти спокойное положение фигуры);
– средне-серый сектор = 3 (положение фигуры кажется напряженным);
– светло-серый сектор = 4 (явно напряженное положение фигуры).

Для оценки стабильности результатов испытуемые повторно проходили эксперимент – через короткий и длительный промежуток времени. Повторные эксперименты выявили существование устойчивых индивидуальных особенностей восприятия полей визуального напряжения экранного кадра у большинства респондентов (рис. 2, 3).

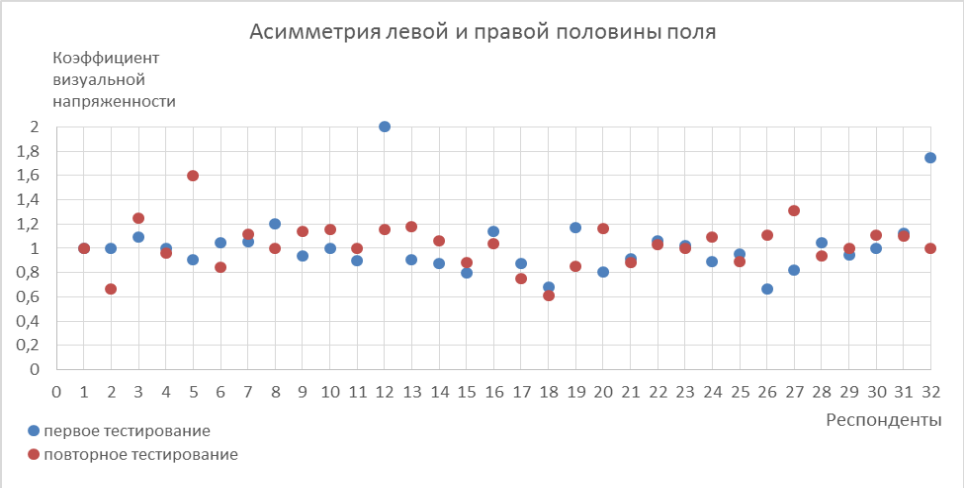


Рис. 2. График сравнения результатов двух тестирований по параметру «асимметрия левой и правой половины поля»

Рис. 3. График сравнения результатов двух тестирований по параметру «асимметрия верхней и нижней половины поля»

При этом индивидуальные особенности восприятия полей визуальной напряженности существенно отличаются от визуализации Арнхейма тем, что:

1. визуальная напряженность объектов в верхней части кадра во многих случаях бывает выше по отношению к объектам, располагающимся в нижней части кадра (асимметрия верх-низ);
2. визуальная напряженность объектов в правой части кадра во многих случаях бывает выше по отношению к объектам, располагающимся в левой части кадра (асимметрия лево-право);
3. визуальная напряженность в центре кадра бывает достаточно высока.

Классификация респондентов по параметрам асимметрии и значениям их коэффициентов представлена на графике (рис. 4).

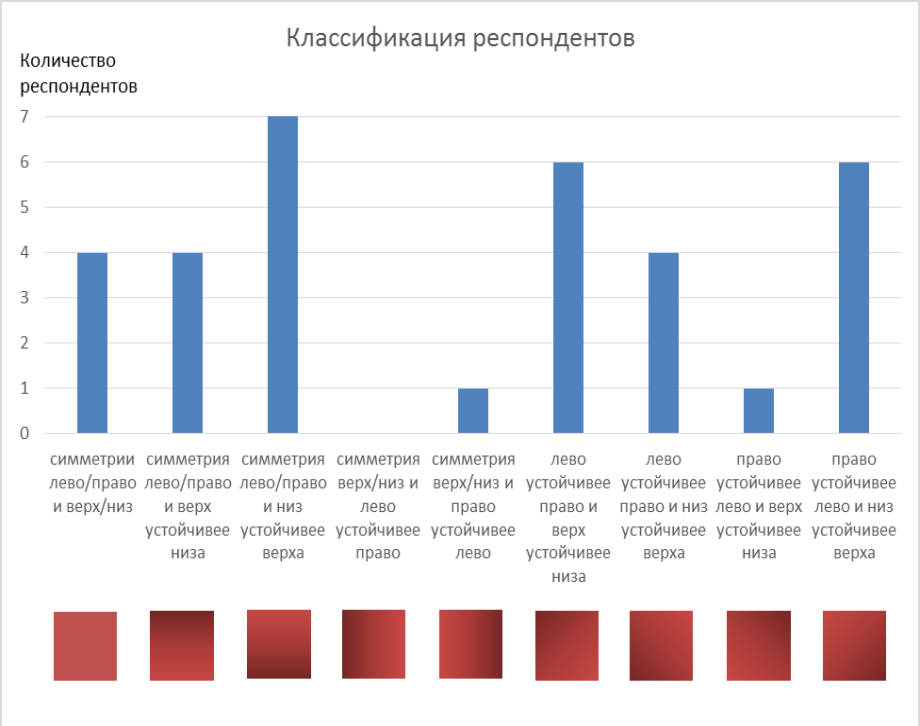


Рис. 4. График распределения респондентов по классам

Причины этих, а также ряда других несоответствий картины поля визуальной напряженности Арнхейма, и полей, полученных в данных экспериментах, можно интерпретировать теоретически на основе анализа различий в индивидуальном визуальном опыте и навыках визуального восприятия респондентов, сформированных в процессе их развития.

Таким образом, получив распределение респондентов по типам восприятия визуальных напряжений компьютерного кадра, можно:

– классифицировать типы организации структуры интерфейса по соответствующим классам респондентов;
– выявить типы восприятия визуальных напряжений компьютерного кадра, для которых не существует соответствующих типов организации структуры интерфейса;
– создать недостающие типы организации структуры интерфейса.

Литература

1. Арнхейм Р. Искусство и визуальное восприятие. – М.: Прогресс, 1974. – 386 с.
2. Design Principles: Visual Weight And Direction [Электронный ресурс]. – Режим доступа: https://www.smashingmagazine.com/2014/12/design-principles-visual-weight-direction/ (дата обращения: 06.02.2018).
3. Smith L.B., Heise D. Perceptual Similarity and Conceptual Structure // Advances in Psychology. – 1992. – V. 93. – P. 233–272.
4. Инишев И.Н. Взаимосвязь визуального восприятия, пространства и действия в герменевтической концепции образа // Вестник ТГПУ. – 2014. – № 7(148). – С. 16–24.

Молодцова Виктория Сергеевна
Год рождения: 1994
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4260
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: VictoriaWD@yandex.ru

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 331.101.5

АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИМЕНЕНИЯ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В ИНФОРМАЦИОННОМ МОДЕЛИРОВАНИИ ЗДАНИЙ (BIM)

Молодцова В.С., Шуклин Д.А.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

В работе проанализированы программное обеспечение технологии дополненной реальности и возможность применения технологии BIM, ее преимущества и сфера внедрения.
Ключевые слова: AR, дополненная реальность, архитектура, информационная модель, BIM, архитектура, строительство.

Понятие «информационная модель» впервые использовал конструктор Чак Истман в одной из своих статей в журнале Американского Института Проектировщиков (AIA). К окончанию 80-х годов прошлого века доктрина получила развитие в Европе и США. Современный термин «Building information modeling» (BIM) – итог соединения британского (Product Information Model) и американского (Building Product Model) вариантов. Он возник в научной работе Роберта Эйша в 1986 году, где были изложены главные принципы нового подхода [1]. Главная мысль ученого заключалась в том, чтобы автоматизировать процесс создания сооружений. Вся нужная информация, в том числе сметы, информационные базы, временные расчеты, объединились воедино в одной компьютерной 3D-модели. Эйш наглядно показал практическую ценность собственной теории, которую в последствии использовал при восстановлении аэропорта Хитроу в столице Великобритании. Это была первая попытка внедрения системы BIM моделирования зданий в мировую архитектурно-строительную деятельность [2]. В 2002 году данная технология получила положительный отклик специалистов всех стран мира и начала активно применяться.

Информационное моделирование BIM проекта позволяет контролировать жизнь сооружения с его закладки до сноса. Возведение – трудоемкий процесс, требующий участия огромного количества профессионалов различных отраслей. BIM-проектирование дает возможность представить их работу как единое целое, высчитать и состыковать все вероятные варианты развития событий, заблаговременно удостовериться, что на этапе проекта не было допущено просчетов, которые могут возникнуть в будущем [3].

Признанные архитекторы и известные строительные компании активно применяют в работе информационные модели. В 2006 году создание музея современных искусств в Колорадо по плану Д. Либескинда доказало, что они ускоряют работу в несколько раз и существенно сокращают затраты. Музей был открыт на год раньше, чем предполагалось, а государственная казна сохранила 230 млн руб. (400 тыс. долл.). В 2008 году один из крупнейших архитекторов современности и лауреат Притцкеровской премии, Фрэнк Гери повторил успех своего коллеги при конструировании Высшей школы музыки в Майами [2].

Все современные строительные планы формируются на компьютере. Специфика данного метода состоит в том, что профессионал трудится не с геометрическими образами, а с цифровой моделью. Информационная модель существует до тех пор, пока есть объекты, которые она воспроизводит. Она модифицируется и модернизируется совместно с сооружениями, вследствие этого иногда ее называют 4D.

Сочетание программного обеспечения (ПО) дополненной реальности (AR, Augmented Reality) с моделями BIM создает новый набор инструментов для экспертов в данной области [4]. Разработка интегрирует цифровые и физиологические характеристики, в итоге чего исключаются промахи, появляется больше гибкости и возможностей в процессе проектирования. В случае необходимости изменений в ходе работы над объектом – вносятся расчеты согласно нормативам. В случае если раньше любые корректировки требовали разработки с нуля, то с поддержкой новых технологий, выполнение анализов, расчетов, корректировок, исключат коллизии и автоматически изменения смет на строительство.

Многие разработчики архитектурных систем автоматизированного проектирования (САПР) и ПО для проектирования зданий используют AR и находят способы применить перспективную технологию в своих программных продуктах.

Градостроители и инженеры в Крайсчерч, Новая Зеландия, использовали приложение CityViewAR впоследствии серии землетрясений. Приложение создано в лаборатории технологий системы «человек-машина» (Human Interface Technology Lab) института Кентербери, и в настоящее время используется для проектирования трехмерной модели зданий в натуральную величину до момента разрушения в результате землетрясения. Это совершенствует работу архитекторов и градостроителей в процессе восстановления.

Приложение SketchUp является собственностью компании Trimble, пользователи которого могут воспользоваться плагином AR-media в сочетании с HoloLens от Microsoft, дабы получить соответствующий навык использования AR, HoloLens разрешает визуализировать трехмерные модели SketchUp в реальной физической среде, куда будут размещены готовые изделия.

Augment. Приложение Augment включает коллекцию трехмерных моделей, но также позволяет загружать и использовать собственные модели и данные. Приложение сканирует двухмерные модели и дает возможность видеть проект в дополненной реальности.

LORAR+. Разработанное австралийской конструкторской фирмой Laing O'Rourke приложение LORAR+ обеспечивает взаимодействие с трехмерными моделями, анимацией, голограммами и виртуальной средой в режиме реального времени. Приложение LORAR+ можно загрузить как с устройств iOS, так и Android.

ViewAR. Сотрудники ViewAR могут по заказу выполнять приложения AR. Они даже могут помочь сформировать библиотеку трехмерных моделей. Приложения имеют много разных характеристик, например, сквозной контроль проектного решения до этапа строительства, способность показывать крупные внешние обновления и реконструкцию так же, как и возможность предварительного просмотра отделки стен обстановки.

BIMevoke – это консалтинговая фирма в области трехмерного проектирования и информационного моделирования зданий. Фирма предлагает бесплатное приложение AR и даже может помочь перевести имеющуюся у вас информацию в трехмерные данные для использования в данном приложении.

SightSpace Pro. Приложение дополненной реальности SightSpace Pro от Limitless Computing позволяет вводить виртуальные проектные решения в реальную среду, обеспечивая возможности сквозного просмотра. Оно предназначено для интеграции с каждой фазой проекта, включая разработку концепции, проектирование, предварительное планирование, строительство и непрерывное техобслуживание. SightSpace Pro доступно для загрузки в App Store и Google Play и может быть использовано с файловыми форматами .DWG, .DXF, .DGN, .SKP, .KMZ, .KML и .DAE [4].

Результат сравнения приложений по таким критериям, как кроссплатформенность, функциональные возможности, поддержка различных языков программирования, тип лицензии приведен в таблице.

Таблица. Анализ программного обеспечения информационного моделирования зданий (BIM)

	Кроссплатформенность	Функциональные возможности	Языки программирования	Тип лицензии
CityViewAR	iOS, Android	Позволяет просматривать трехмерные модели зданий в натуральную величину, как они выглядели до разрушений	C#, Java	Boost Software License
SketchUp	iOS, Android	Визуализация трехмерных моделей в реальной физической среде	C/C + +, Python, Java	Free + Commercial SDK option
Augment	Android	Включает коллекции общедоступных трехмерных моделей и данных, позволяя также использовать собственные модели и данные	C#, Java	Free + Commercial SDK option
LORAR +	iOS	Обеспечивает взаимодействие с трехмерными моделями, анимацией, голограммами и виртуальной средой в режиме реального времени	C/C + +, Python, Java, Ruby, Lua	Free + Commercial SDK option
ViewAR	Android	Способность показывать крупные внешние обновления и реконструкцию, так же как и возможности предварительного просмотра отделки стен обстановки	Java, Objective-C	Free + Commercial SDK option
BIMevoke	iOS, Android	Может помочь перевести имеющуюся информацию в трехмерные данные для использования в данном приложении	C#, Java	Boost Software License
SightSpace Pro	iOS, Android	Позволяет вводить виртуальные проектные решения в реальную среду, обеспечивая возможности сквозного просмотра	C + +, Java, Objective-C, .Net	Free + Commercial SDK option

Таким образом, важным преимуществом внедрения BIM-технологии дополненной реальности в строительство и архитектуру – это наглядность модели объекта, возможность

визуальной оценки объемно-планировочных, инженерных и других смежных решений, а также выявление и оперативное устранение погрешностей еще на этапе проектирования с использованием специального программного обеспечения.

Литература

1. Дополненная реальность – это просто [Электронный ресурс]. – Режим доступа: http://habrahabr.ru/post/144280/ (дата обращения: 04.01.2019).
2. Миловидов С.В. Принципы «дополненной реальности» и интерактивная реконструкция в строительстве [Электронный ресурс]. – Режим доступа: http://artculturestudies.sias.ru/vypusk-6-2013/sotsialnaya-filosofiya-i-sotsiologiya/848.html (дата обращения: 04.01.2019).
3. Кравцов А.А. Использование технологии дополненной реальности для визуализации виртуального объекта в архитектуре [Электронный ресурс]. – Режим доступа: http://ej.kubagro.ru/2012/10/pdf/54.pdf (дата обращения: 04.01.2019).
4. Туменова С.А., Варквасова Т.Ч. Рынок дополненной и виртуальной реальности: состояние и перспективы развития // Известия Кабардино-Балкарского научного центра РАН. – 2017. – № 3(77). – С. 88–93.

Мосолова Ирина Сергеевна
Год рождения: 1997
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P34702
Направление подготовки: 09.04.02 – Мультимедиа-технологии
в искусстве театра, кино и телевидения
e-mail: lira62960@gmail.com

Смолин Артем Александрович
Год рождения: 1977
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.философ.н.
e-mail: smolin@corp.ifmo.ru

УДК 004.946

МЕТОДЫ ВОВЛЕЧЕНИЯ ПОЛЬЗОВАТЕЛЕЙ В ВИРТУАЛЬНУЮ РЕАЛЬНОСТЬ

Мосолова И.С.
Научный руководитель – к.философ.н. Смолин А.А.

В работе рассмотрена проблема вовлеченности в VR-контент при использовании гарнитуры виртуальной реальности. Поднимаются вопросы: о возникновении чувства присутствия и вовлеченности; о том, какие факторы виртуальной реальности могут повлиять на достижение этого эффекта; как психология восприятия человека способствует появлению чувства «потока». На примерах из сферы медиакультуры показано как теоретические аспекты преобразуются в практические методы вовлечения.

Ключевые слова: виртуальная реальность, вовлечение, иммерсивность, восприятие виртуальной среды, состояние потока.

Начиная с 2012 года – сразу после выхода очков виртуальной реальности Oculus, специалисты прогнозировали виртуальное будущее и повсеместное использование гарнитур к 2020 году. Однако до сих пор сложно представить, что очки «придут» в каждый дом. Согласно интервью, проведенного студентами Университета ИТМО в ноябре 2018 года, 70% респондентов, в возрасте от 18 до 25 лет, имели опыт просмотра VR-контента. И только один человек из опрошенных хотел бы посмотреть контент второй раз, остальные же отвечали отрицательно, ссылаясь на плохое качество, отсутствие чувства погружения и интересной механики, способной удержать интерес пользователя. Очевидно, что проблема в несоответствии реальных и ожидаемых уровней качества представляемого контента. Зритель чувствует себя не комфортно в предложенных обстоятельствах, ему не хватает ментальной вовлеченности в процесс. Однако можно предположить, что развитие интерактивного сюжета в зависимости от действий зрителя при просмотре VR-контента увеличивает вовлеченность в продукт и послужит причиной его дальнейшего качественного развития.

По мнению американского ученого Уильяма Дугласа, [1] виртуальная реальность (VR) – это «иммерсивная и интерактивная имитация реалистичных и вымышленных сред», в которой человек может взаимодействовать с данным миром и получать ответные реакции на свои действия. С точки зрения исследования методов вовлечения, наибольший интерес у авторов вызывает взаимодействие, при котором возможно перемещение внутри среды, но нет

активного интерактива. Ведь такая среда позволяет сделать больше, чем грамотно срежиссированный ролик.

Рассматривая современные гарнитуры, стоит отметить, что большинство поддерживает лишь два чувствительных канала: визуальный и звуковой, и именно интерактивность позволяет говорить о иммерсивности или чувстве присутствия, которое пользователь ощущает в той или иной мере [2]. В такой среде человек играет важную роль, его действия непосредственно влияют на содержание всей виртуальной среды, что в итоге повышает вовлеченность в происходящее. Однако наличие этих факторов в среде вовсе не гарантируют, что пользователь увлечется процессом, ведь речь идет об индивидуальной психологии восприятия. Разберемся, какая именно взаимосвязь стоит между чувством вовлеченности, увлеченности и процессом восприятия. Увлеченность, как процесс концентрации человека на определенной идее, более осознанное и направленное чувство, которое возникает индивидуально и непредсказуемо, вызывая интерес к какому-либо аспекту. Психическая вовлеченность же понятие более универсальное, определяющееся состоянием, при котором индивид осознано и неосознанно общается со стимулами окружающей среды и изменяет его. Разбирая структуру чувства вовлеченности, психолог Михай Чиксентмихайи [3] выделяет состояние потока – «интенсивную включенность в эмпирическую спонтанную деятельность, которая может быть материальной или умственной». В такие моменты внимание сфокусировано только на конкретной преследуемой задаче, что позволяет личности полностью раскрыться. Поток имеет прямую связь с сознанием и психической энергией и способствует появлению позитивного настроя и уходом от стрессовых ситуаций и негативных мыслей.

Чтобы достичь состояния потока необходимо выполнение таких условий как: четкость сознания (определение структуры и целей последующей деятельности с оперативной обратной связью, которые позволяют выбрать наиболее подходящий способ выполнения необходимых задач); адаптация проблемной ситуации под качества личности смотрящего. Ситуация будет интересна человеку только в том случае, если он обладает компетенциями для ее решения. Наибольший интерес будут вызывать: трудная проблема, находящаяся в зоне ближайшего развития пользователя; слияние сознания с действием (возникновение чувства увлеченности и пониженная реакция на внешние и внутренние раздражители); чувство контроля (осознание наличия навыков, как инструментов воздействия на ситуацию); ускорение времени за счет уникальности переживаемого опыта; получение удовольствия от процесса.

Современная позитивная психология говорит о том, что формирование здоровой положительной реакции в основном зависит от наличия потоковых переживаний, ведь все они переводят человека на новую ступень восприятия, расширяют горизонты его способностей [4]. Именно поэтому описанные выше особенности восприятия так важно учитывать при выявлении конкретных методов вовлечения пользователей из отдельных проектов. Например, фильм «Генри» (Henry – 2015 год, Oculus story studio) сделан специально под формат VR и удостоен международного признания наградой Emmy. В нем пользователю доступно перемещение по локации, но в то же время возможности реагировать на происходящее нет. Зато такая возможность есть у главного героя – он реагирует на положение пользователя и смотрит ему прямо в глаза. Это неожиданная ответная реакция позволяет вовлечь смотрящих в историю от начала до конца. Создатели фильма «Вторжение» (Invasion! – 2016 год, Baobab studios) отмечают, что их успех основан на эмоциональной достоверности продукта. Несмотря на ограниченность пользователя в действиях, сопереживание достигается благодаря грамотной режиссуре. Главный герой здесь выступает проводником в историю, адаптируя ее под зрителя. Рекламный ролик «Merrell» (2015 год, Framestore VR) для производителей туристической обуви демонстрирует, как слияние сознания с действием может вовлекать зрителей. В ролике пользователю предстоит пройти горную тропу, и ключевым фактором

здесь является наличие таких же реальных декораций, которые дают достоверную информацию осязательному каналу.

Рассматривая виртуальную анимацию и фильмы-360, нельзя не обратиться к классическим полнометражным проектам. В этой области набирает обороты интерактивный сюжет. В настоящий момент к такому формату прибегают для наибольшего погружения в проблему, как, например, в российском социальном фильме Н. Покровской «Все сложно» про ВИЧ-положительную девушку, или для поддержания сюжетной линии, как в популярном сериале «Черное зеркало». Несмотря на разные сюжеты фильмы похожи в одном – тут зритель выступает как активный создатель истории, от выбора которого зависит дальнейшие события.

В индустрии развлечений компьютерные игры занимают лидирующие позиции по степени вовлечения пользователей в продукт. С точки зрения виртуальной реальности наибольший интерес здесь вызывают игры с открытым миром, так как поведение пользователя в таком случае близко к свободному перемещению в VR. В таких играх, как «The Witcher 3» и «Minecraft», пользователей привлекает продуманный нелинейный сюжет, где каждое решение имеет вес и изменяет ход в будущей игры. Четкость сознания и чувство контроля вводит игроков в состояние потока. Не имея практически никаких навыков в игре можно достичь больших высот, т.е. ключевой фактор популярности – адаптация и постепенное восхождение по состоянию потока.

Таким образом, можно сделать вывод, что наиболее эффективные методы вовлечения пользователя в виртуальную реальность могут быть основаны на принципах визуального контакта с героем, на следование персонажа за пользователем, на нелинейном ходе сюжета и на адаптацию способностей пользователя к уровню текущего развития. Эти интерактивные подходы наряду с новаторскими алгоритмами используются при создании прототипа продукта для проведения эксперимента в рамках научно-исследовательской работы.

Литература

1. Douglas Y., Hargadon A. The Pleasure Principle: Immersion, Engagement, Flow // Proceedings of the Elevent ACM Conference on Hypertext and Hypermedia. – 2000). – P. 153–160.
2. Кузьмина А.С. Виртуальная реальность как средство безопасного контакта с травмирующей реальностью в психотерапии [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/n/virtualnaya-realnost-kak-sredstvo-bezopasnogo-kontakta-s-travmiruyuschey-realnostyu-v-psihoterapii (дата обращения: 25.12.2019).
3. Чиксентмихайи М. Поток: Психология оптимального переживания / Пер. с англ. – 3-е изд. – М.: Смысл: Альпина нон-фикшн, 2011. – 461 с.
4. Величковский Б.Б. Психологические факторы возникновения чувства присутствия в виртуальных средах [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/n/psihologicheskie-faktory-vozniknoveniya-chuvstva-prisutstviya-v-virtualnyh-sredah (дата обращения: 12.01.2019).

Никитинская Вера Михайловна
Год рождения: 1994
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4262
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: nikitinskaya.v@gmail.com

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.04

КЛАССИФИКАЦИЯ ИНСТРУМЕНТОВ ГЕНЕРАЦИИ СТАТИЧЕСКИХ ВЕБ-САЙТОВ

Никитинская В.М., Государев И.Б.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

Работа выполнена в рамках темы НИР № 615892 «Исследование и разработка в области информационных технологий».

В работе приведено описание основных особенностей генераторов статических веб-сайтов. Выделен ряд характеристик инструментов генерации и приведена их классификация по ряду укрупненных групп.
Ключевые слова: статический веб-сайт, генераторы статических сайтов.

В настоящее время существует ряд способов представления информации в интернете: интерактивные веб-приложения, информационные сайты, сайты-брошюры, платформы электронной коммерции, блоги и форумы, посадочные страницы. В целом все данное разнообразие веб-проектов может быть разделено на две крупные группы по способу организации хранения и представления файлов веб-сайта: статические и динамические сайты [1]. Статический веб-сайт представляет собой набор файлов, содержащих HTML-разметку, CSS-стили и код клиентской логики на языке JavaScript. Данные файлы размещены на веб-сервере и по запросу со стороны клиента передаются «как есть», так как они заранее сгенерированы. Соответственно, по запросу пользователя на стороне сервера не требуются выполнение задач серверной логики, обращение к базе данных и сборка страниц из шаблонов и полученных данных. Отсутствие данного шага значительно ускоряет загрузку страниц, обеспечивает надежность в условиях резко возрастающей нагрузки (например, при непредвиденном росте интереса к странице со стороны пользователей), а также безопасность данных для пользователей и владельцев сайта [2].

Несмотря на то, что простейший статический сайт может быть написан «вручную» и размещен «как есть», при наличии на сайте двух и более страниц или страниц с повторяющимися элементами (например, панель меню и нижняя информационная панель) становится целесообразным разделение шаблонов страниц сайта и данных, заполняющих данные шаблоны. Генератор статического сайта – это инструмент, используемый

самостоятельно либо включаемый как один из этапов в систему непрерывной сборки/интеграции, и позволяющий автоматизировать процесс размещения данных в подготовленных шаблонах [3]. Существуют разнообразные генераторы, однако, их принцип работы сводится к следующим этапам [4]:

1. считывание и размещение в отведенных слотах шаблонов элементов данных сайта;
2. переименование файлов и размещение в них дополнительной информации с учетом указанных метаданных;
3. конвертация полученных текстовых файлов в формат HTML;
4. необязательно – транспиляция и компиляция файлов стилей и скриптов в формат CSS и JS;
5. размещение полученных файлов в папке с готовой сборкой сайта.

Генераторы статических сайтов, как правило, не имеют графического интерфейса: управление осуществляется с помощью интерфейса командной строки, что делает данные инструменты ориентированными на технически подготовленных пользователей. Однако набор команд, предоставляемый программой-генератором, при наличии достаточной документации позволяет осуществлять настройку сборки и работать с сайтом без значительных затруднений. При этом многие генераторы статических сайтов, помимо указанной выше базовой функциональности, предоставляют расширенный набор возможностей, что позволяет подобрать такой инструмент под нужды различных проектов и с учетом разнообразных особенностей разработки.

Генераторы статических сайтов обладают рядом базовых характеристик, обуславливающих особенности работы с ними [5]. Для каждого из таких генераторов выделяют один или несколько предпочтительных сценариев использования. Язык программирования, на котором написан сам генератор, определяет требуемую экосистему для его использования: так, генератор, созданный на языке JavaScript, потребует установки на компьютере разработчика среды Node.js. Также генераторы различаются по тем языкам разметки, метаданных и данных, которые они поддерживают в процессе генерации. Генераторы могут предоставлять набор плагинов и тем оформления, расширяющих возможности генерации сайтов. Наконец, могут быть предоставлены и другие дополнительные инструменты разработки. Соответственно, может быть приведена следующая классификация инструментов генерации с учетом данных характеристик, где каждая из них представляет собой отдельный класс. Внутри данного класса выделяют две или более категории, к которым можно отнести различные генераторы.

Класс «Предпочтительные сценарии использования» включает в себя следующие:

1. генераторы блогов. Как правило, генераторы сайтов данного типа включают в себя расширенные возможности управления метаданными;
2. генераторы маркетинговых сайтов или посадочных страниц. Предоставляются очень гибкие шаблоны и инструменты частичной шаблонной разметки (partials);
3. генераторы документации содержат плагины для оформления примеров программного кода;
4. генераторы портфолио, личных страниц включают дополнительные, гибко редактируемые темы визуального оформления;
5. генераторы платформ электронной коммерции дополнительно содержат систему управления контентом сайта (Content Management System, CMS);
6. инструменты публикации книг и других электронных изданий. Зачастую генератор статических сайтов в данной категории представляет собой веб-сервис, имеет интеграцию с платформами размещения электронных изданий.

Как правило, один или несколько предпочтительных сценариев использования указываются непосредственно в документации к генератору статических сайтов разработчиками данного инструмента. Однако следование данной рекомендации не является обязательным, остается на усмотрение пользователя, поэтому выбор инструмента следует

вести с учетом наибольшего возможного ряда характеристик генератора и требований к итоговому продукту.

Класс «Язык и экосистема» разделяется на две категории:

1. генераторы, написанные на языке JavaScript;
2. генераторы на любых других языках.

Инструменты из первой категории, как правило, распространяются в виде npm-пакета. Таким же образом к ним подключаются любые дополнительные модули и темы оформления. Соответственно, работа с данными генераторами требует установки среды Node.js и менеджера зависимостей npm. Часто такие генераторы обладают гибкой системой транспиляции и сборки, поскольку используют систему Webpack. Возможно написание шаблонов с использованием таких распространенных фреймворков, как Vue и React.

Инструменты из второй категории могут обладать особенностями, обусловленными конкретным языком и экосистемой: так, генераторы на языке Go предлагают преимущество крайне высокой скорости сборки сайта. Генерацию сайта можно встроить в виде этапа сборки проекта, написанного на соответствующем языке. Однако при этом необходимо специальным образом настраивать окружение, менеджер пакетов для конкретной среды разработки и прочие инструменты.

В классе «Языки разметки, шаблонов, метаданных и данных» могут быть выделены следующие две категории:

1. генераторы, использующие наиболее широко распространенный набор языков: Markdown как язык разметки, JSX или Liquid – язык шаблонов, YAML для разметки метаданных и JSON для данных сайта;
2. генераторы, использующие специализированные наборы языков: AsciiDoc и Textile для разметки; ERB, Go, Jinja, EJS, Haml, Handlebars, Nunjacks – шаблонизаторы; использование сторонних API или GraphQL для управления метаданными и данными.

Класс «Подключаемые модули и плагины» включает в себя спектр возможных наборов и способов кастомизации генератора статических сайтов. Можно выделить две укрупненные категории:

1. инструменты, изначально предоставляющие максимальный набор функций: GitBook, Middleman, Hexo; наименее настраиваемые, однако, готовые для использования и имеющие широкий ряд готовых тем оформления и подключаемых модулей;
2. инструменты, содержащие минимальный базовый набор функций и дающие пользователю возможность редактировать функциональность под нужды конкретного проекта. Так, генератор Metalsmith представляет собой «обертку» для запуска плагинов и предполагает, что вся функциональность формируется пользователем самостоятельно.

Класс «Дополнительные инструменты разработки» содержит в себе ряд категорий, к которым можно относить инструменты генерации статических сайтов исходя из наличия или отсутствия у них рассматриваемых функций и инструментов:

1. генераторы, имеющие интерфейс командной строки;
2. генераторы, содержащие встроенный локальный сервер;
3. инструменты, предоставляющие функцию «горячей перезагрузки» страниц при разработке;
4. генераторы, использующие встроенные инструменты препроцессинга материалов сайта: систему Webpack или собственные системы сборки;
5. генераторы, имеющие функцию code splitting для подгрузки только используемых модулей;
6. генераторы, позволяющие осуществлять импорт из систем Wordpress или RSS-фида;
7. генераторы, имеющие дополнительные функции на основе интеграции с хостингом статических сайтов, например, GitHub или Netlify.

Широкое разнообразие генераторов статических сайтов и обширная функциональность, предоставляемая ими, делает статические сайты удобным и современным средством публикации контента в сети Интернет. Представленная классификация может служить основой для первичного выбора инструмента генерации сайта исходя из нужд конкретного

веб-проекта, а также базой для проведения дальнейших исследований по теме разработки статических веб-сайтов.

Литература

1. Классификация сайтов [Электронный ресурс]. – Режим доступа: https://www.internet-technologies.ru/articles/klassifikaciya-saytov-vidy-saytov.html (дата обращения: 04.01.2019).
2. Camden R., Rinaldi B. Working with Static Sites. – O'Reilly Media, Inc., 2016. – 195 p.
3. Boucas E. An Introduction to Static Site Generators [Электронный ресурс]. – Режим доступа: https://davidwalsh.name/introduction-static-site-generators (дата обращения: 04.01.2019).
4. Rinaldi B. Static Site Generators. – O'Reilly Media, Inc., 2015. – 57 p.
5. A List of Static Site Generators for JAMstack Sites [Электронный ресурс]. – Режим доступа: https://www.staticgen.com/ (дата обращения: 04.01.2019).

Носков Илья Игоревич
Год рождения: 1991
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: noskovii@mail.ru

Богатырев Владимир Анатольевич
Университет ИТМО, факультет программной инженерии и компьютерной техники, д.т.н., профессор
e-mail: vladimir.bogatyrev@gmail.com

УДК 004.94

ОБОСНОВАНИЕ ВЫБОРА СРЕДЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДЛЯ ИССЛЕДОВАНИЯ ОТКАЗОУСТОЙЧИВЫХ КОМПЬЮТЕРНЫХ СЕТЕЙ

Носков И.И.
Научный руководитель – д.т.н., профессор Богатырев В.А.

Работа выполнена в рамках темы НИР № 619296 «Разработка методов создания и внедрения киберфизических систем».

Проведено обоснование выбора среды имитационного моделирования для исследования отказоустойчивых компьютерных сетей и дана классификация существующих сред моделирования компьютерных сетей. Приведены критерии выбора среды моделирования, позволяющие исследовать поведение отказоустойчивых компьютерных сетей. Выполнена классификация сетевых протоколов, обеспечивающих надежность современных сетей передачи данных. Дано описание среды моделирования OMNeT++, которая позволяет строить и исследовать отказоустойчивые компьютерные сети.

Ключевые слова: имитационное моделирование, компьютерные сети, отказоустойчивость, надежность, OMNeT++.

Введение. В настоящее время наблюдается большой рост компьютерных сетей, позволяющих передавать данные на огромных скоростях, доходящих до нескольких терабит в секунду на магистральных узлах [1]. Даже при кратковременном сбое, вызванного деструктивным воздействием на канал (случайным или целенаправленным), в такой сети, произойдет потеря огромного количества данных [2–4]. Также возрастает количество сервисов с высоким уровнем доступности и систем реального времени, которые должны предоставлять свои услуги без перебоя в течение длительного времени [5]. Для обеспечения отказоустойчивости компьютерных сетей необходимы методики, позволяющие строить надежные сетевые топологии с резервированием сетевого оборудования, использующие протоколы обеспечения сетевой отказоустойчивости в своей работе. В настоящее время основным инструментом исследования компьютерных сетей без создания физических топологий является имитационное моделирование.

Моделирование компьютерных сетей предоставляет большие возможности для разработки и исследования сетей различной топологии и размера. Использование

имитационного моделирования в процессе проектирования компьютерных сетей позволяет пользователю решать задачи по их оптимизации и управлению, а также выполнять апробацию различных решений, не затрачивая при этом средств на создание реальной топологии [6]. В настоящее время разработано существенное количество сред моделирования различного назначения, отличающихся специализацией и наличием возможностей по моделированию той или иной предметной области. При выборе среды моделирования для исследования отказоустойчивых компьютерных сетей, пользователю будут доступны различные библиотеки и специализированные модули, позволяющие эффективно строить модели, наиболее точно отражающие реальный объект исследования.

Таким образом, выбор специализированной среды имитационного моделирования для исследования отказоустойчивых компьютерных сетей являлся актуальной задачей, выполнение которой позволяет детально проектировать и исследовать надежные сетевые конфигурации.

Классификация методов обеспечения сетевой отказоустойчивости. Обеспечение сетевой отказоустойчивости в компьютерных сетях различного назначения достигается за счет резервирования активного сетевого оборудования и (или) каналов связи, а также путем использования сетевых протоколов различного уровня. С точки зрения базовой архитектуры IP-сетей все активное сетевое оборудование работает на канальном и сетевом уровнях модели OSI, и именно они определяют надежность и отказоустойчивость компьютерных сетей в целом [7]. На рис. 1 приведена классификационная схема протоколов обеспечения отказоустойчивости компьютерных сетей на канальном и сетевом уровнях.

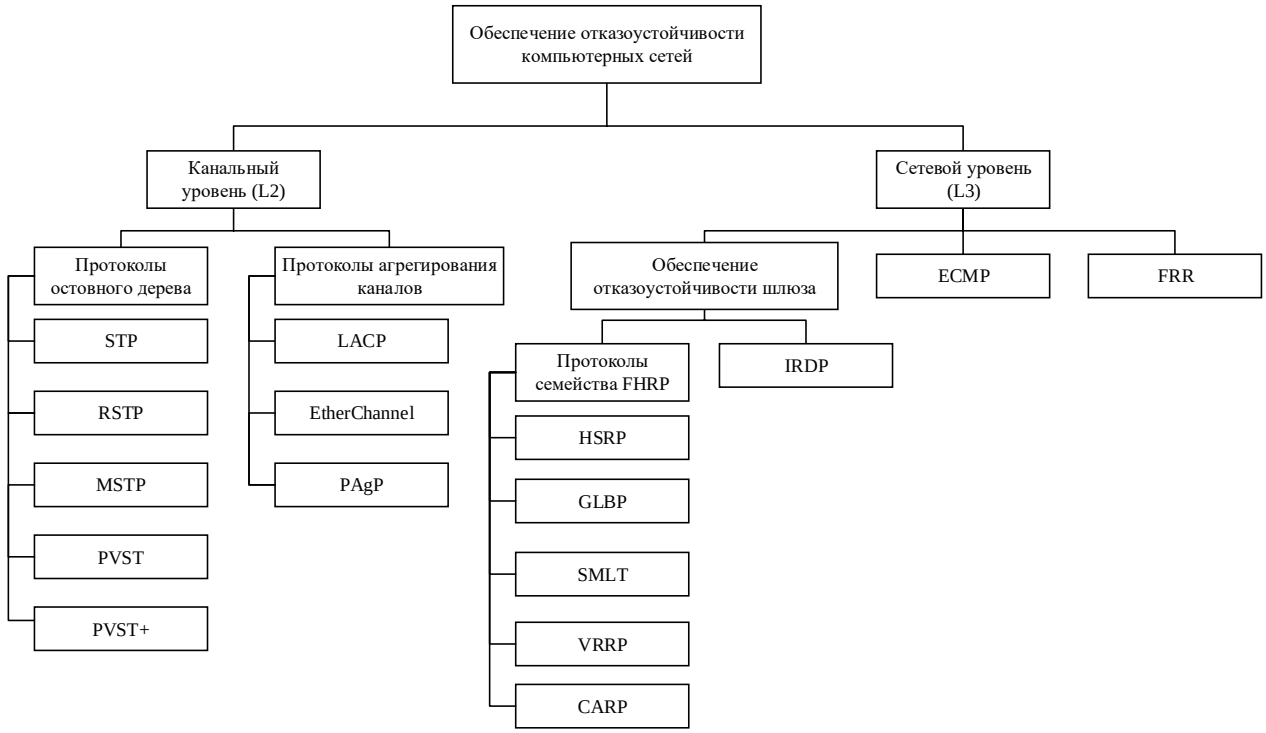


Рис. 1. Классификация протоколов, обеспечивающих отказоустойчивость компьютерных сетей

Как видно из рис. 1, протоколы обеспечения отказоустойчивости на канальном уровне делятся на две большие группы – протоколы остовного дерева и протоколы агрегирования каналов [8]. Протоколы сетевого уровня, в свою очередь, делятся на протоколы, обеспечивающие отказоустойчивость шлюза (протоколы семейства FHRP и IRDP), ECMP и FRR для MPLS-сетей.

При выборе среды имитационного моделирования для исследования отказоустойчивых компьютерных сетей необходимо учитывать возможность использования в моделях

рассмотренных выше сетевых протоколов. Реализация данных протоколов самостоятельно в среде моделирования является трудоемким процессом и отнимет много времени от непосредственного исследования сетевых конфигураций.

Аналитический обзор сред моделирования. Перед непосредственным выбором и аналитическим обзором существующих сред моделирования необходимо было выделить основные критерии, по которым будет осуществляться данный выбор [9]. Для этого выделены ключевые характеристики и возможности сред, позволяющих моделировать работу отказоустойчивых компьютерных сетей:

1. специализированная среда моделирования компьютерных сетей (необходима для более точного представления предметной области и большей адекватности разрабатываемых моделей);
2. поддержка средой имитационного моделирования возможности распределенного моделирования для исследования сложных и высоконагруженных топологий компьютерных сетей (запуск исполнения модели на нескольких вычислительных узлах) [10];
3. наличие большой библиотеки разработанных моделей и модулей, предоставляющих функционал сетевых протоколов различного уровня (в том числе протоколов обеспечения сетевой отказоустойчивости);
4. возможность бесплатного использования данной среды в академических целях;
5. наличие графического интерфейса пользователя для наглядного представления разрабатываемых моделей компьютерных сетей.

На рынке программного обеспечения средств моделирования доступно достаточное количество разноплановых пакетов программ. Существуют специальные ориентированные на моделирование вычислительных сетей программные системы, в которых процесс создания модели упрощен. Такие программные системы сами генерируют модель сети на основе исходных данных о ее топологии и используемых протоколах, об интенсивностях потоков запросов между компьютерами сети, протяженности линий связи, о типах используемого оборудования и приложений. Программные системы моделирования могут быть узкоспециализированными и достаточно универсальными, позволяющими имитировать сети самых различных типов.

Если вести речь о многофункциональных программных пакетах, то наибольшее распространение имеют следующие продукты:

– GloMoSim / QualNet;
– OMNeT++;
– OPNET Modeler;
– NS-3.

Все перечисленные программные продукты обладают схожими наборами средств моделирования компьютерных сетей, так как они развиваются довольно давно, библиотеки готовых моделей покрывает почти все существующие протоколы, используемые в компьютерных сетях. В них осуществлена поддержка создания распределенных и параллельных моделей для моделирования сетей, включающих сотни и тысячи рабочих станций. Главным отличием этих сред является способ создания модели сети. В NS-3 модель создается при помощи программы, написанной на C++ или OTcl, в то время как QualNet и OPNET Modeler обладают удобным графическим интерфейсом, а среда OMNeT++ позволяет создавать модели как с помощью языка C++, так и с помощью графического интерфейса пользователя. Еще одно отличие – полнофункциональное использование QualNet и OPNET Modeler возможно только с приобретением достаточно дорогих лицензий, в отличие от свободно распространяемых сред NS-3 и OMNeT++.

В таблице приведены результаты сравнения описанных выше специализированных сред имитационного моделирования.

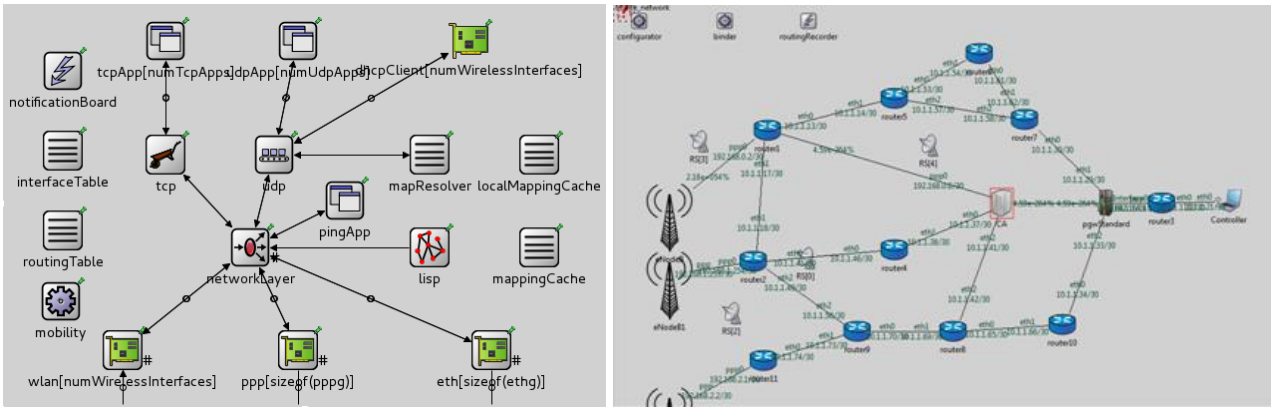
Таблица. Сравнение специализированных сред имитационного моделирования

Название	Поддержка распределенного моделирования	Возможность бесплатного использования	Богатая библиотека разработанных моделей и модулей	Наличие графического интерфейса пользователя
QualNet	+	–	–	+
OMNeT++	+	+	+	+
OPNET Modeler	+	–	+	+
NS-3	+	+	+	–

Из полученных результатов сравнения видно, что среда моделирования OMNeT++ является наиболее подходящей для исследования отказоустойчивых компьютерных сетей.

Среда моделирования OMNeT++. Среда OMNeT++ является кроссплатформенной, имеет богатую библиотеку сетевых компонентов и реализаций протоколов на языке C++, которые можно менять или разрабатывать свои, имеет графический интерфейс, а также подробную англоязычную документацию. Так как исходный код данной среды открыт, она динамично развивается, часто выходят обновления и новые библиотеки модулей и классов.

Среда моделирования OMNeT++ позволяет строить модели сетевых протоколов и устройств, используя при этом принцип композиции и усложнения моделей, выделения их в отдельный блок (рис. 2, а), а затем используя эти блоки при построении более сложных моделей (рис. 2, б).



а б

Рис. 2. Модель маршрутизатора (а); схема сети с использованием моделей маршрутизаторов (б)

Разработка моделей и их конфигурирование выполняется с помощью языка программирования C++ и встроенного языка описания сети NED, представляющим собой графический способ построения модели.

В настоящее время для системы моделирования OMNeT++ разрабатывается много различных компонентов и протоколов программистами со всего мира. Наборы компонентов и протоколов объединяются в библиотеки, которые называются фреймворками. В настоящее время основным фреймворком для OMNeT++ является INET Framework, который поддерживается сообществом разработчиком OMNeT++. INET Framework является библиотекой с открытым исходным кодом, которая предоставляет модели проводных, беспроводных и мобильных сетей.

Наличие такого большого количества готовых компонентов позволяет пользователю не заниматься реализацией сетевых протоколов согласно документам, RFC с помощью языка C++, а сосредоточиться на создании сетевых моделей и проведении экспериментов для дальнейших исследований.

Фреймворк ANSA-INET расширяет рассмотренную ранее библиотеку INET в части разработки протоколов для проводных сетей. ANSA-INET также является проектом с открытым исходным кодом. Данный фреймоворк предоставляет следующие основные возможности/улучшения относительно INET Framework: протоколы семейства FHRP (HSRP, VRRPv2, GLBP), алгоритмы динамической маршрутизации (IS-IS, RIPv2, RIPng, EIGRP, Babel), протоколы управления L2-уровнем (CDP, LLDP), протоколы мультикаст маршрутизации (PIM-DM, PIM-SM, IGMPv2, IGMPv3) и др. [11]. Наличие моделей протоколов семейства FHRP является большим преимуществом OMNeT++ в исследовании отказоустойчивости на сетевом уровне.

После создания модели компьютерной сети и задания всех настроек пользователь может запустить выполнение моделирования в интерактивном сеансе и наблюдать весь процесс работы сети в реальном времени. По окончании эксперимента пользователю становятся доступны результаты в виде графиков и наборов данных с полной трассировкой событий моделирования.

Заключение. В результате проделанной работы проведена классификация методов и протоколов обеспечения сетевой отказоустойчивости компьютерных сетей. Выделены основные критерии сред моделирования компьютерных сетей, на основе которых был произведен аналитический обзор существующих средств.

Среда моделирования OMNeT++ является наиболее подходящей для исследования и изучения отказоустойчивых компьютерных сетей, благодаря большой гибкости, кроссплатформенности, богатой библиотеке классов, написанных на языке программирования C++, а также своими средствами проведения экспериментов и анализа полученных результатов.

Среда OMNeT++ использовалась для исследования компьютерной сети с агрегированием каналов со случайным методом доступа при резервировании передач [12]. Дальнейшая работа и исследования на основе имитационных экспериментов в данной среде будут направлены на разработку методики и выработку рекомендаций по построению отказоустойчивых компьютерных сетей.

Литература

1. Шувалов В.П., Егунов М.М., Минина Е.А. Обеспечение показателей надежности телекоммуникационных систем и сетей. – М.: Горячая линия – Телеком, 2015. – 168 с.
2. Богатырев В.А., Богатырев С.В. Надежность мультикластерных систем с перераспределением потоков запросов // Изв. вузов. Приборостроение. – 2017. – Т. 60. – № 2. – С. 171–177.
3. Богатырев В.А., Богатырев С.В., Богатырев А.В. Надежность кластерных вычислительных систем с дублированными связями серверов и устройств хранения // Информационные технологии. – 2013. – № 2. – С. 27–32.
4. Богатырев В.А., Богатырев А.В., Голубев И.Ю., Богатырев С.В. Оптимизация распределения запросов между кластерами отказоустойчивой вычислительной системы // Научно-технический вестник информационных технологий, механики и оптики. – 2013. – № 3(85). – С. 77–82.
5. Носков И.И. Протоколы обеспечения сетевой отказоустойчивости информационных экономических систем // Информационно-технологическое обеспечение цифровой экономики: сборник статей. – 2018. – С. 67–73.
6. Кельтон В., Лоу А. Имитационное моделирование. – СПб.: Питер; Киев: Изд-я группа BHV, 2004. – 847 с.
7. Носков И.И. Методы и алгоритмы обеспечения надежности и отказоустойчивости компьютерных сетей // Синергия Наук. – 2018. – № 24. – С. 603–616.

8. Богатырев В.А., Богатырев С.В., Богатырев А.В. Оптимизация древовидной сети с резервированием коммутационных узлов и связей // Телекоммуникации. – 2013. – № 2. – С. 42–48.
9. Богатырев В.А., Кармановский Н.С., Попцова Н.А., Паршутина С.А., Богатырев С.В. Имитационная модель поддержки проектирования инфокоммуникационных резервированных систем // Научно-технический вестник информационных технологий, механики и оптики. – 2016. – Т. 16. – № 5. – С. 831–838.
10. Fujimoto R.M. Parallel and Distributed Simulation Systems. – Wiley Interscience, 2000. – 195 p.
11. ANSAINET extends INET framework for OMNeT++ // [Электронный ресурс]. – Режим доступа: https://ansa.omnetpp.org/ (дата обращения: 04.01.2019).
12. Носков И.И., Богатырев В.А., Сластихин И.А. Имитационная модель локальной компьютерной сети с агрегированием каналов и случайным методом доступа при резервировании передач // Научно-технический вестник информационных технологий, механики и оптики. – 2018. – Т. 18. – № 6(118). – С. 1047–1053.

Павлов Евгений Игоревич
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41622
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: quemsaurose@gmail.com

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004

ПРОЕКТИРОВАНИЕ ОБРАЗОВАТЕЛЬНОЙ ВЕБ-ПЛАТФОРМЫ «ЦИФРОВАЯ ГРАМОТНОСТЬ»

Павлов Е.И.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

В работе обобщены отечественные и зарубежные исследования, посвященные теме определения цифровой грамотности, выявлено общее представление о необходимости развития цифровых компетенций в контексте цифрового потенциала Российской Федерации, проанализированы преимущества и недостатки образовательных платформ как средств обучения.
Ключевые слова: цифровая грамотность, онлайн-образование, образовательная платформа.

Мы живем в мире, наполненном технологиями. Всего десять лет назад мы бы не столкнулись с таким количеством компьютеров, сколько существует сейчас. Реальность такова, что члены общества больше не могут позволить себе быть технофобами. Цифровая грамотность является одним из ключевых компонентов цифрового гражданина – человека, который использует информационные технологии для взаимодействия с окружающим миром. Ведь за последние несколько десятилетий современная действительность превратилась в цифровую среду. Почти все, что люди делают сегодня, требует своего рода цифровых знаний или грамотности: от серфинга в Интернете до поиска работы. Компьютерная грамотность сегодня – это не просто особый навык для определенного круга людей. Именно это заставляет людей принимать нарастающие темпы технологического развития и готовиться к прогрессивному будущему.

Цифровая грамотность стала частью нашей жизни. Люди легко ориентируются в Интернете, ищут статьи и видео, обмениваются информацией с друзьями. Эти задачи кажутся легкими, однако они не так просты для всех. Есть еще много людей, для которых проблема цифровой безграмотности стоит достаточно остро. В связи с этим необходимо повышать цифровую грамотность всех людей, чтобы они могли комфортно жить в современном цифровом мире.

Перевод информации в цифровую форму стал доминантным в развитии межличностных коммуникаций, профессиональной деятельности практически любого направления, образования, получения и предоставления услуг (в том числе государственных), отдыха и многого другого. В настоящее время цифровые технологии

используются не только как высокотехнологичные средства коммуникации, но и как инструмент для образования и работы [1].

Важно понимать, что цифровая трансформация общества получила бурный скачок благодаря стремительному развитию информационных технологий и Интернета. По причине того, что веб проник во все сферы деятельности человека, сейчас практически любой сервис фактически является веб-ресурсом или веб-приложением.

Наряду с понятием цифровой грамотности широко используется ряд родственных понятий: медиаграмотность, информационная, компьютерная, сетевая и интернет-грамотность. Их возникновение было вызвано социокультурными изменениями, обусловленными расширением возможностей применения средств информационно-коммуникационных технологий в различных сферах жизнедеятельности общества.

Под цифровой грамотностью понимается способность находить, оценивать, перерабатывать и создавать контент с использованием информационных технологий и Интернета.

Новизна понятия «цифровая грамотность» привела к тому, что многие люди по своему его комментируют и разрабатывают различные критерии для измерения цифровой грамотности. Несмотря на это, не существует конкретного мнения о том, какие метрики использовать, причем разные авторы предоставляют разные критерии, некоторые из которых слишком расплывчаты или иным образом слабо определены.

Определение «цифровая грамотность» содержит три составляющих компонента [2]:

1. навыки и способности, которые позволяют эффективно использовать цифровые технологии в повседневной жизни – цифровые компетенции. Например, умение использовать онлайн-сервисы для получения товаров и услуг, а также выполнять финансовые операции через Интернет;
2. цифровое потребление – применение цифровых компетенций в рамках определенных жизненных ситуаций, что приводит к использованию различных цифровых ресурсов;
3. цифровая безопасность, которая заключается во владении навыками безопасной работы в сети как технического, так и социального характера: способность защитить свои персональные данные, обеспечить конфиденциальность и целостность информации, обезопасить ее от компьютерных вирусов. Компетентность населения в области цифровой безопасности является необходимым условием безрискового поведения в глобальном цифровом пространстве и основой успешной реализации цифровой грамотности.

Цифровая грамотность включает в себя знание о СМИ, возможность изучать и понимать смысл входящих цифровых сообщений, оценивать их достоверность и качество, навыки синтеза информации и совместной работы, реакции на «цифровые события».

Цифровая грамотность находит применение во многих сферах человеческой жизнедеятельности:

1. образование. Образовательные учреждения постоянно обновляют свои учебные программы, чтобы соответствовать непрерывно ускоряющимся темпам технологического развития общества. Это заключается в использовании компьютеров в процессе обучения, использование онлайн-курсов и распространение учебных материалов через Интернет;
2. общество. Цифровая грамотность необходима для правильного использования различных цифровых платформ. Грамотность в социальных сетях и мессенджерах помогает людям оставаться в контакте с другими людьми, получать актуальную информацию, и даже покупать и продавать товары и услуги;
3. трудовая деятельность. В настоящее время многие профессии требуют знания компьютера и Интернета для выполнения основных задач. Все офисные рабочие места оборудованы компьютерами, портативными устройствами и соответствующей инфраструктурой локальной сети.

По исследовательским данным [3], в рамках развития цифровой грамотности необходимо проводить мероприятия, нацеленные на совершенствование знаний и умений

граждан по эффективному и безопасному использованию Интернета и цифровых технологий, а также разработать образовательные программы в сфере цифровой грамотности и разместить их на открытых порталах электронного образования.

В 2017 году в России принята комплексная государственная программа «Цифровая экономика Российской Федерации», которая определила основные направления цифровой трансформации российского общества.

Важный момент, который отражен в программе – это повышение цифровой грамотности населения и ввод аттестационных нормативов.

Текущее положение Российской Федерации (РФ) на мировом цифровом рынке характеризуется значительным отставанием от мировых лидеров по уровню использования цифровых технологий в повседневной деятельности потребителей и компании.

С 2015 года Региональная общественная организация «Центр Интернет-технологий» (РОЦИТ) занимается исследованием уровня цифровой грамотности населения РФ в качестве комплексного показателя, который отражает степень зрелости общества в применении цифровых технологий и использовании ресурсов Интернета.

Результаты трех ежегодных исследований свидетельствуют о достаточно низком уровне индекса цифровой грамотности как в целом по стране, так и по федеральным округам. Однако в целом прослеживается положительная динамика [4].

По итогам мероприятий, подлежащих реализации по направлению «Кадры и образование» в рамках программы «Цифровая экономика Российской Федерации», предполагается работа системы повышения грамотности населения в сфере информационной безопасности, медиапотребления и использования интернет-сервисов к IV кварталу 2019 г., которая позволит ежегодно увеличивать долю граждан, повысивших грамотность в данной сфере.

В качестве результата работы данного направления предполагается обеспечение вхождения России «на лидирующие позиции международных рейтингов в области оценки математической и цифровой грамотности населения» и разработка методики по обеспечению цифровой грамотности широких слоев населения, включая старшее поколение.

Образовательная система в России в настоящий момент претерпевает стремительные изменения. Огромная популярность технологий дистанционного образования привела к тому, что для многих людей обучение становится непрерывным на протяжении всей жизни. Так как современные цифровые технологии проникают во все сферы деятельности человека, в том числе образование, отдельные участки образовательного процесса могут быть автоматизированы.

В связи с этим все чаще разрабатываются специализированные онлайн-сервисы, которые позволяют пользователям изучать учебные материалы, обмениваться информацией и выполнять задания. Их преимущество заключается в том, что через асинхронное обучение (такой метод обучения, при котором контакт между обучающимся и учителем задержан во времени) ускоряется процесс передачи информации обучающимся.

В настоящее время термин «образовательная платформа» уверенно вошло в жизнь всех сотрудников образовательной сферы. Конкретное определение этого термина пока не устоялось, но опираясь на научную работу [5], можно сказать, что образовательная платформа – это личностно ориентированный интернет-ресурс, посвященный вопросам образования и развития, содержащий учебные материалы и предоставляющий их пользователям.

Параллельно с этим получают распространение новые формы образования, например, проблемно-ориентированная форма, которая подразумевает использование реальных проблем в качестве стимула для того, чтобы учиться. Анализ этих проблем приводит к приобретению новых знаний по дисциплине и навыков решения проблем.

Имеющаяся потребность на образовательные услуги диктует, что обучение должно давать возможность обучаться быстро, понятно и частично (или полностью) дистанционно, вестись на реальных примерах, а также предусматривать многоуровневую концепцию

обучения, которая будет мотивировать учащегося совершенствовать имеющиеся или приобретать новые компетенции.

Особую значимость образовательная платформа приобретает для повышения знаний в сфере цифровой грамотности. Например, курс, традиционно состоящий из тематических семинаров и лекций, в рамках вышеописанного проблемно-ориентированного подхода должен превратиться в пошаговый алгоритм успешного становления человека в современном мире в качестве цифрового гражданина.

Использование веб-платформы в качестве средства обучения позволит обучающимся получать материал в любое время и с любой периодичностью. Более того, обучающийся может легко вновь изучить пропущенный им материал без необходимости присутствовать в учебном заведении.

Рассматривая все вышесказанное, можно заключить, что в настоящее время необходимость повышения цифровой грамотности уже ни у кого не вызывает сомнений. Можно сделать вывод, что изучение исследований в сфере цифровой грамотности и образования является целесообразным и имеет практическую значимость в виде реализации программы повышения навыков цифровой грамотности на учебной платформе. Руководствуясь предложенной концепцией эта платформа может быть охарактеризована как образовательная веб-платформа, которая, используя технологии смешанного обучения, организует процесс обучения по дисциплине «Цифровая грамотность».

Литература

1. Цифровое будущее. Каталог навыков медиа- и информационной грамотности. – М.: Межрегиональный центр библиотечного сотрудничества (МЦБС), 2013. – 68 с.
2. Цифровая грамотность [Электронный ресурс]. – Режим доступа: http://цифроваяграмотность.рф (дата обращения: 04.01.2019).
3. Мак Р.И., Мак Н.И. Основные направления деятельности по повышению уровня цифровой грамотности населения // Актуальные вопросы права, экономики и управления: сборник. – 2018. – С. 20–22.
4. Индекс 2017 – Цифровая грамотность [Электронный ресурс]. – Режим доступа: http://цифроваяграмотность.рф/mindex/2017/ (дата обращения: 04.01.2019).
5. Тиунова Н.Н. Образовательные платформы как средство интенсификации профессиональной подготовки студентов колледжа // Профессиональное образование в России и за рубежом. – 2016. – № 2(22). – С. 103–108.

Паршин Дмитрий Сергеевич
Год рождения: 1992
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P41622
Направление подготовки: 09.04.04 – Программная инженерия
e-mail: dima1parshin@gmail.com

Учанева Анастасия Алексеевна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P41622
Направление подготовки: 09.04.04 – Программная инженерия
e-mail: nastya.uchaneva@gmail.com

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004.55

ОСОБЕННОСТИ ВЕРСТКИ ВЕБ-САЙТОВ ПОД МОБИЛЬНЫЕ УСТРОЙСТВА

Паршин Д.С., Учанева А.А.
Научные руководители: к.педагог.н., доцент Государев И.Б.;
к.педагог.н., доцент Шуклин Д.А.

В работе рассмотрены различные особенности создания веб-сайтов для мобильных платформ, направленные на адаптацию под различные размеры дисплеев и ускорение загрузки. Приведен сравнительный анализ CSS-фреймворков.
Ключевые слова: адаптивный дизайн, отзывчивая верстка, CSS-фреймворки, Flexbox, Grid CSS.

Создание веб-сайтов под мобильные устройства предполагает использование различных техник, определяющих поведение элементов на странице. Важным аспектом перед началом создания сайта является выбор метода верстки. В работе дано описание основных методов верстки. Фиксированная верстка задается путем задания фиксированной ширины блоков в пикселях, которые не меняют своего размера [1]. Плюсы: скорость и простота создания сайта. Минусы: отсутствие адаптации под размер экрана.

При использовании «резиновой» верстки ширина блоков задается в процентах, поэтому элементы растягиваются при изменении размера дисплея в заданных пропорциях. Плюсы: отображение элементов сайта в заданных пропорциях и размерах вне зависимости от размера дисплея. Минусы: при большом разбросе размеров экранов сложно реализовать дизайн так, чтобы при любом разрешении все элементы выглядели одинаково хорошо, чтобы блоки не «разъезжались», а картинки и видео не вылезали за границы блоков [2].

Адаптивная верстка была предложена Ароном Густафсоном в работе [3]. Используя этот метод верстки, разработчики создают несколько независимых вариантов дизайна для разных устройств. Когда пользователь заходит на сайт, сервер определяет тип устройства и разрешение экрана и «отдает» ту или иную версию сайта. При этом каждая версия сайта заранее оптимизирована под то или иное разрешение экрана. Стили переключаются от одного breakpoint (точки изменения разрешения экрана) к другому, т.е. мы имеем фиксированные макеты, например, для iPhon-ов и iPad-ов, не затрагивая устройства с иными разрешениями экрана [4]. Плюсы: адаптация под популярные разрешения устройств, возможность создать «мобильную» версию сайта. Минусы: ориентация на определенные разрешения экранов.

«Отзывчивая» (responsive) верстка. Метод предложен Итаном Маркоттом, объединяющий в себе «резиновую», адаптивную верстки и гибкие изображения. «Отзывчивый» дизайн автоматически подстраивается под любую ширину экрана, как и резиновый, при этом в процессе подстройки может менять расположение блоков, размер картинок, могут пропадать или изменяться некоторые элементы (например, обычное горизонтальное меню может поменяться на выпадающее).

Создание мобильной версии сайта. Разработчик верстает несколько версий сайта, как правило, с урезанным функционалом под различные мобильные устройства (табл. 1).

Таблица 1. Сравнительный анализ технологий верстки

	Фиксированная	«Резиновая»	Адаптивная	Responsive
Изменение размера блока при изменении размера viewport	Нет	Да	Да	Да
Изменение размера пропорционально размеру viewport	Нет	Да	Нет	Да
Предлагает отличающиеся версии композиции сайта в зависимости от viewport	Нет	Нет	Да	Да
Плавность изменения размеров блоков	Нет	Да	Нет	Да

В результате сравнительного анализа было выявлено значительное преимущество подхода Responsive верстки.

Flexbox – модуль макета гибкого контейнера. Он предоставляет более гибкий и функциональный инструмент, но имеет ограничения при использовании с устаревшими версиями браузеров (от IE11), которые можно обойти при использовании autoprefixer. Flexbox разметка предоставляет один из наиболее эффективных способов выравнивания, позиционирования и распределения места внутри контейнера между элементами, даже если их размер динамичен (поэтому эта технология имеет в своем название английское слово Flex, что в переводе от flexible – значит гибкий, уступчивый) и заранее не известен [5]. Основная задача Flexbox – предоставление возможности изменения своих элементов по высоте и ширине, чтобы они максимально вмещались в доступное место родительского контейнера, в том числе в тех случаях, когда необходимо соответствовать всем типам дисплеев устройств. Flex-контейнер расширяет вложенные в него элементы с целью заполнения доступного пространства и также урезает их, чтобы избежать переполнения. Особенностью Flexbox

является то, что он независим от направления, в отличие от inline модели, которая основана на горизонтальном размещении и блочной, основанной на вертикальном размещении. Свойства модуля могут применяться к контейнеру (родительскому элементу) и к отдельным элементам этого контейнера. Элементы в Flexbox имеют два потока, располагаются вдоль главной оси (main axis), которая по умолчанию имеет горизонтальную направленность, или же вдоль поперечной (cross axis).

CSS Grid – это набирающая популярность модель шаблона, адаптированная для двумерных шаблонов. В отличие от flexbox работает в двух измерениях. Представляет собой набор пересекающихся горизонтальных и вертикальных линий, которые создают размеры и позиционируют систему координат для элементов в Grid-контейнере. «Гриды» включают в себя интуитивно понятный «ASCII-графический» синтаксис, в котором возможно виртуально видеть шаблон в коде, это облегчает использование Grid -модели.

Таблица 2. Сравнительный анализ технологий разметки Flexbox и Grid

	Flexbox	Grid
Позиционирование	Одномерное	Двумерное
Первично	Контент	Макет
Поддержка технологии в процентах глобального трафика	92%	85%

В результате сравнительного анализа были выявлены отличия технологий разметки Flexbox и Grid (табл. 2).

Для ускорения процесса разработки адаптивных веб-сайтов, исключения множества ошибок верстки и упрощения работы разработчика были созданы CSS-фреймворки. В табл. 3 рассмотрены плюсы и минусы некоторых самых популярных CSS-фреймворков: Bootstrap 4, Zurb Foundation, Pure CSS, Materialize, Gumby, Foundation.

Таблица 3. Сравнительный анализ CSS-фреймворков

CSS-фреймворк	+	–
Bootstrap 4	1. Легкая настройка 2. Много различных функций 3. Классическая сетка (12 столбцов) 4. Поддерживает множество различных тем оформления и подключение дополнительных плагинов 5. Поддержка jQuery 6. JS-скрипты 7. Поддерживает масштабирование на всех устройствах 8. Широко применяется практически для любой CMS (Joomla, Bitrix) 9. Совместимость с браузерами	избыточный код
Zurb Foundation	1. Легкая настройка 2. Много различных функций, свойств	Конфликты со сторонними скриптами
Pure CSS	1. Размер всех файлов данного фремворка равен 3,8 KB, что позволяет веб-страницам мгновенно загрузиться. 2. Гибкость (24-колоночная сетка)	1. Мало функций 2. Не используются JS-файлы
Materialize	1. Много различных функций 2. Классическая сетка (12 столбцов)	Конфликты с плагином JavaScript
Gumby	Существует возможность индивидуально настраивать сетку	Избыточный код

CSS-фреймворк	+	–
Foundation	1. Широкий набор готовых предустановок для сайта 2. Совместимость с jQuery, но при этом заменили эту библиотеку на Zepto.js 3. Бесплатен как для личного использования, так и для организаций	Сложность в освоении фреймворка новичками

Выбор фреймворка зависит от поставленной задачи, ориентации на мобильное или десктопное устройство.

Методы уменьшения времени загрузки сайта:

1. GZIP – реализуемый на стороне сервера метод, основанный на двух алгоритмах: DEFLTATE и Хаффмана. Первый заменяет повторные вхождения данных на «ссылки», второй является методом кодирования с переменной длиной, которая присваивает более короткие коды к наиболее часто повторяющимся «символам». Позволяет уменьшить HTML, CSS и JS-файлы на 70%;
2. кэширование. При использовании кэширования браузер сохраняет часть файлов у пользователя. При первом посещении сайта загрузка происходит в обычном режиме, в том время как при последующих используются уже загруженные на устройство пользователя файлы;
3. использование «спрайтов». Большое значение в скорости загрузки страницы имеет количество запросов от клиента к серверу. Уменьшить количество запросов поможет использования «спрайтов» – однотипных изображений (например, иконок), объединенных в одно. Также если у изображений сходная палитра, то объединенное изображение будет меньше по размеру, чем совокупность исходных файлов;
4. удаление лишних фрагментов. Используемые на сайте файлы могут содержать ни разу не вызываемые элементы, которые, тем не менее, пользователь загружает. Примером могут служить шрифты с не используемыми языками или элементами. Избавившись от них, можно уменьшить количество загружаемой информации.

Заключение. Таким образом, современный веб-сайт стоит создавать с учетом актуальных технологий, позволяющих увеличить скорость загрузки и адаптацию под разнообразные экраны дисплеев мобильных устройств.

Литература

1. Плохенко В.Г., Симоневич Е.И. Применением современных стандартов HTML5 и CSS при проектировании сайта // Международный журнал экспериментального образования. – 2016. – № 9-2. – С. 321–322.
2. Лещев Д. Создание интерактивного web-сайта: учебный курс. – СПб.: Питер, 2003. – 544 с.
3. Gustafson A. Adaptive web design: crafting rich experiences with progressive enhancement. – 2nd ed. – New Riders, 2015. – 264 p.
4. De Bra P. Design issues in adaptive web-site development [Электронный ресурс]. – Режим доступа: http://wwwis.win.tue.nl/~debra/asum99/debra/debra.html (дата обращения: 04.01.2019).
5. Tanoue A. FlexBox: Sandboxing internet servers based on layer-7 contexts // Proceedings – IEEE Symposium on Computers and Communications. – 2008. – P. 386–391.

Перес Ортис Мигель Анхель
Год рождения: 1980
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P4214
Направление подготовки: 09.04.01 – Информатика и вычислительная техника
e-mail: miguelangel.miguelperez@gmail.com

УДК 004.94

ВИЗУАЛИЗАЦИЯ И СЕГМЕНТАЦИЯ ТРОПИЧЕСКИХ ЦИКЛОНОВ В СПУТНИКОВЫХ ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ ТЕХНИКИ ДВОРЖАКА

Перес О.М.А.

Работа выполнена в рамках темы НИР № 617028 «Информационные системы и измерительные технологии для защиты окружающей среды, техносферной и экологической безопасности».

Спутниковые изображения являются очень полезным инструментом в ежедневном прогнозировании и разработке исследований. Метеорологические явления изучаются и обнаруживаются по наблюдениям метеорологических спутников. Данная работа направлена на реализацию, визуализацию и сегментацию тропических циклонов с использованием технологии Дворжака, разработку алгоритмов для будущей идентификации облачных моделей по всей траектории циклона.
Ключевые слова: циклоны, спутник, Дворжак.

Введение. Распознавание шаблонов также называется чтением, идентификацией и распознаванием форм. Шаблоны являются производными от процессов сегментации, извлечения характеристик и описания, где каждый объект представлен набором дескрипторов. Система распознавания должна назначать каждому объекту свою собственную категорию или класс (набор объектов, которые разделяют некоторые характеристики, которые отличают их от других) [1–5].

1. Чтобы распознать закономерности, необходимы следующие процессы:
   – сбор данных;
   – особенности извлечения;
   – принятие решений.
2. Тропический циклон – это метеорологический термин, который используется для обозначения турбулентной системы, характеризующейся замкнутой циркуляцией вокруг центра низкого давления и созданием сильных ветров и проливных дождей. Тропические циклоны черпают энергию из влажного конденсированного воздуха, создавая сильные ветры. Они отличаются от других циклонических штормов, таких как полярные минимумы, тепловым механизмом, который питает и превращает их в штормовые системы «теплого сердца».
3. Метод Дворжака – это метод получения оценок интенсивности тропических циклонов по спутниковым снимкам. Верн Дворжак разработал систему распознавания образов с использованием дерева решений в начале 1970-х годов.
4. С помощью спутниковых снимков можно узнать текущее состояние атмосферы, определить и отследить крупные метеорологические учреждения, такие как холодные фронты, ураганы, волны и другие. В данной работе предложена разработка с использованием библиотечных алгоритмов OpenCV-Python, визуализации и сегментации спутниковых изображений в качестве метода наблюдения за тропическим циклоном, что позволяет специалистам и исследователям из Метеорологического института Кубы лучше прогнозировать формирование и развитие этого шторма, используя технику Дворжака в качестве руководства.

Обзор предметной области. Метеорологические спутники являются одним из наиболее часто используемых инструментов наблюдения за атмосферой в службе прогнозирования Метеорологического института Кубы. В этом исследовании предложено изучение тропических циклонов с использованием техники Дворжака (рис. 1), для которой в облаке реализованы алгоритмы распознавания образов, что обеспечивает большую эффективность в изучении этого природного явления.

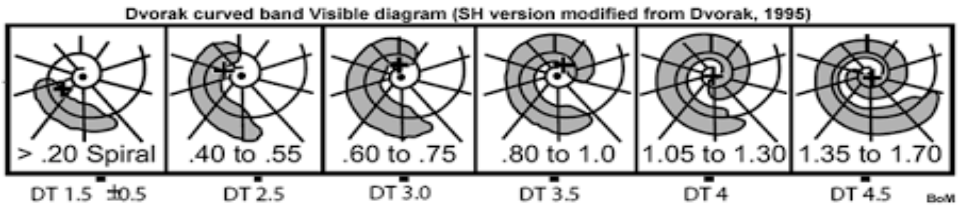


Рис. 1. Таблица для определения категории тропических циклонов

Содержание исследования. Алгоритмы:

1. фильтры, пороги;
2. морфология;
3. контуры.

Алгоритмы библиотеки OpenCV-Python реализованы для выполнения сегментации в спутниковых изображениях. В первом решении используется ядро Гаусса и пороговое значение изображения, используемого в качестве пороговой функции, использование метода бинарных расчетов Оцу. Во втором решении выполняется морфология, а в третьем решении можно сделать хорошую визуализацию контуров облаков.

На рис. 2–7 показана сегментация шторма категорий 1–6 каждый день.

```
1)img = cv2.imread(img,0)
# глобальное пороговое значение
ret1, th1 = cv2.threshold(img, 200, 255, cv2.THRESH_BINARY)
# Пороговое значение Оцу
ret2, th2 = cv2.threshold(img,2000,400, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
# Пороговое значение Оцу после гауссовой фильтрации
blur = cv2.GaussianBlur(img, (5,5),100)
ret3, th3 = cv2.threshold(blur,4000,800, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
# построить все изображения и их гистограммы
images = [img, 0, th1,
          img, 0, th2,
          blur, 0, th3]
titles = ['цикл. день 1','Гистограмма','бинарный (v=200)',
          'цикл.день 6','Гистограмма',"бинарный+ОЦУ",
          'фильтр по Гауссу','Гистограмма',"Гауссу+бинарный+ОЦУ"]
for i in range (3):
   plt. subplot (3,3, i*3+1), plt. imshow(images[i*3],'gray')
   plt. title(titles[i*3]), plt. Xticks ([]), plt. Yticks ([])
   plt. Subplot (3,3, i*3+2), plt. hist(images[i*3]. Ravel (),256)
   plt. title(titles[i*3+1]), plt. Xticks ([]), plt. Yticks ([])
   plt. Subplot (3,3, i*3+3), plt. imshow(images[i*3+2],'gray')
   plt. title(titles[i*3+2]), plt. Xticks ([]), plt. Yticks ([])
```

```
2)  # Загрузить изображение в сером масштабе
img = cv2.imread('img', 0)
# Создать желаемое ядро
kernelType = 1
if kernelType == 0:
   kernel = np. ones ((5, 5), np. uint8)
elif kernelType == 1:
   kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (5, 5))
elif kernelType == 1:
   kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5))
elif kernelType == 2:
   kernel = cv2.getStructuringElement(cv2.MORPH_CROSS, (5, 5))
# Применить различные преобразования
erosion = cv2.erode(img, kernel=kernel, iterations=1)
dilation = cv2.dilate(img, kernel=kernel, iterations=1)
opening = cv2.morphologyEx(img, op=cv2.MORPH_OPEN, kernel=kernel)
closing = cv2.morphologyEx(img, op=cv2.MORPH_CLOSE, kernel=kernel)
gradient = cv2.morphologyEx(img, op=cv2.MORPH_GRADIENT, kernel=kernel)
tophat = cv2.morphologyEx(img, op=cv2.MORPH_TOPHAT, kernel=kernel)
blackhat = cv2.morphologyEx(img, op=cv2.MORPH_BLACKHAT, kernel=kernel)
# Показать Результаты
titles = ['циклон день 1', 'эрозия', 'дилатация', 'открытие', 'закрытие', 'уклон', 'цилиндр', 'черная шляпа']
images = [img, erosion, dilation, opening, closing, gradient, tophat, blackhat]
for i in range(len(titles)):
   plt. subplot (3, 3, i + 1)
   plt. imshow(images[i], cmap='gray', interpolation='bicubic')
   plt. title(titles[i])
plt. xticks ([])
 plt. yticks ([])
```

```
3)im = cv2.imread('img')
imgray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
img = cv2.imread('img',0)
ret, thresh = cv2.threshold(img, 0,255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
image, contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
img = cv2.drawContours(img, contours, -1, (255,255,255), 3)
plt. Figure (fig size= (10,10))
plt. subplot (1,2,1), plt. title ('Original Image'), plt. imshow(im)#,'red')
plt. subplot (1,2,2), plt. title('findContours'), plt. imshow (img, 'gray') #,'red')
Количество контуров
print ('количество обнаруженных контуров: ', len(contours))22
```

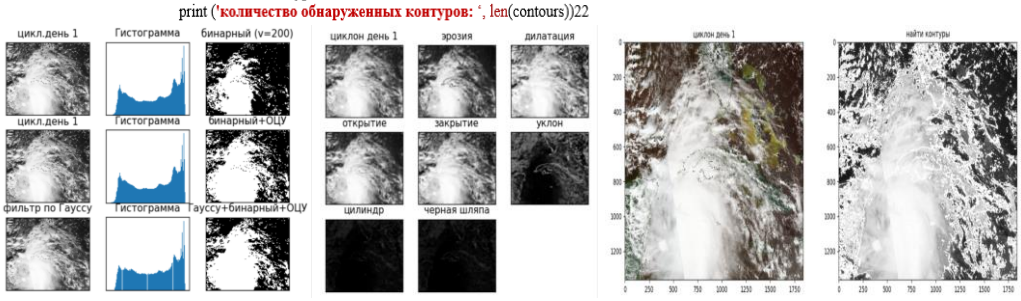


Рис. 2. Получены результаты: 1 день, техническая категория Дворжака 1, количество обнаруженных контуров: 11514

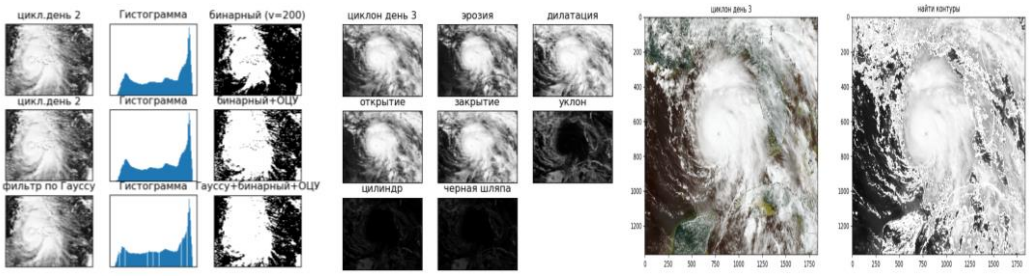


Рис. 3. Получены результаты: 2 день, техническая категория Дворжака 2-3, количество обнаруженных контуров: 9904

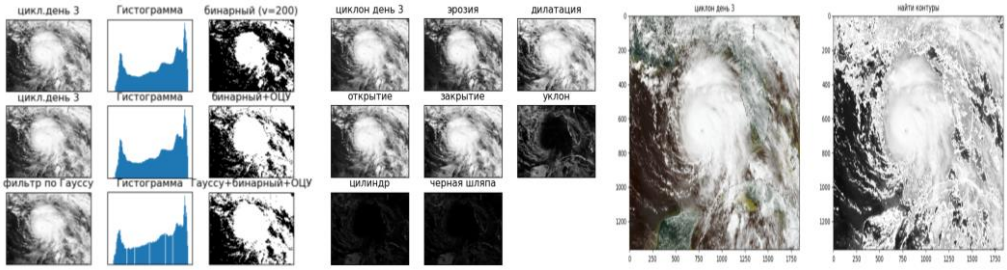


Рис. 4. Получены результаты: 3 день, техническая категория Дворжака 3, количество обнаруженных контуров: 11949

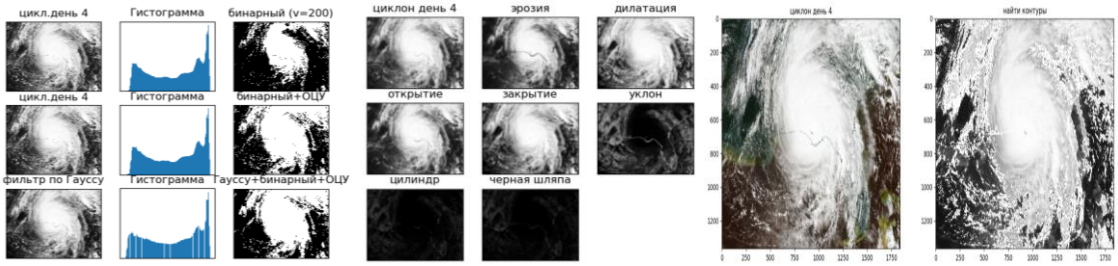


Рис. 5. Получены результаты: 4 день, техническая категория Дворжака 4, количество обнаруженных контуров: 11991

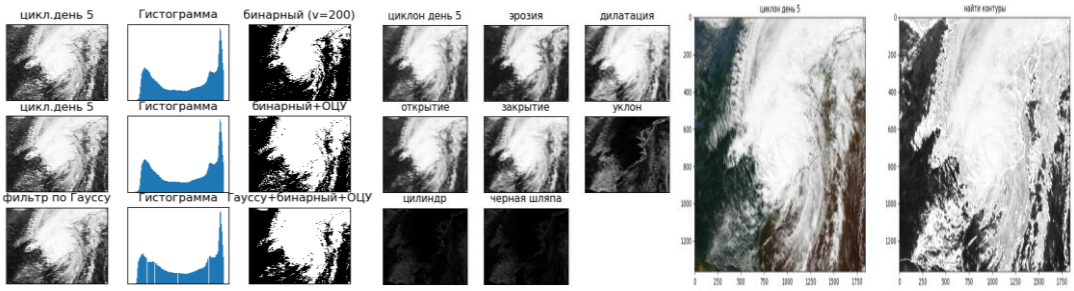


Рис. 6. Получены результаты: 5 день, техническая категория Дворжака 5, количество обнаруженных контуров: 12760

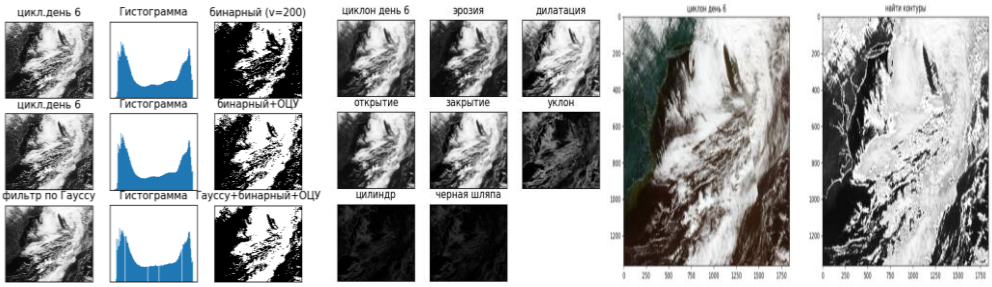


Рис. 7. Получены результаты: 6 день, техническая категория Дворжака 6, количество обнаруженных контуров: 13184

Заключение. Распознавание облаков на спутниковых изображениях является темой, которая в последнее время рассматривается как очень важный инструмент для идентификации

и распознавания метеорологических явлений, в связи с этим большое значение имеет использование алгоритмов для достижения лучшей сегментации на этих изображениях. Алгоритмы машинного обучения, предложенные в работе, требуют глубокого анализа для представления облачных моделей в тропических циклонах с использованием техники Дворжака, поэтому предлагается продолжить максимизацию каждого направления на самом высоком уровне и добиться полной реализации и визуализации в веб-среде предлагаемой системы.

Литература

1. Ernesto A. Meyer. Glosario de términos técnicos, Entrada P, «pattern recognition». – 1995.
2. Achara T., Ray A.K. Image Processing Principles and Applications. – Arizona: Wiley, 2005. – 451 p.
3. Tropical Cyclone Forecasters Reference Guide 2 [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/235030150_Tropical_Cyclone_Forecasters_Reference_Guide_2_Tropical_Climatology (дата обращения: 04.01.2019).
4. Dvorak V.F. A technique for the analysis and forecasting of tropical cyclone intensities from satellite pictures. NOAA Technical Memorandum NESS 45 (Revision of NOAA TM NESS 36). – 1973. – 19 p.
5. The OpenCV Reference Manual. – 2016. – 929 p.

Постникова Полина Игоревна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р41601
Направление подготовки: 09.04.02 – Компьютерная графика и веб-дизайн
e-mail: kentucky.anisimova@yandex.ru

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004.51

ИССЛЕДОВАНИЕ ПОЛОЖИТЕЛЬНЫХ И ОТРИЦАТЕЛЬНЫХ СТОРОН МОБИЛЬНЫХ ПРИЛОЖЕНИЙ КАК ИНСТРУМЕНТОВ КОММУНИКАЦИИ (НА ПРИМЕРЕ ПРИЛОЖЕНИЙ «ГОСУСЛУГИ», «СБЕРБАНК ОНЛАЙН» И «НАШ САНКТ-ПЕТЕРБУРГ)»

Постникова П.И.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

В работе выявлены основные достоинства и недостатки мобильных приложений как средств коммуникации с целью использования полученных результатов в дальнейших исследованиях. Помимо этого, даны и рекомендации к улучшению мобильных приложений на основе анализа отзывов к приложениям «Госуслуги», «Сбербанк Онлайн» и «Наш Санкт-Петербург».
Ключевые слова: коммуникации, приложения для коммуникаций, мобильные приложения, интерфейс, дизайн интерфейсов.

В современном мире популярность мобильных устройств растет с каждым днем. По данным аналитиков к 2018 году количество пользователей мобильных приложений для коммуникации во всем мире превышает 50% всех пользователей мобильных устройств. Речь идет не только о повседневном общении, в котором люди часто используют приложения, но и о корпоративных коммуникациях – как между сотрудниками компании, так и компании с потребителями.

Мобильное приложение можно определить как программный продукт, создаваемый для мобильных устройств и направленный на удовлетворение потребностей пользователей [1]. Они охватывают большой спектр услуг в разных сферах деятельности и предоставляют широкие возможности для взаимодействия.

Мобильные приложения становятся все более популярными и чаще используются различными организациями в качестве инструмента коммуникации. Это связано с тем, что мобильные приложения являются достаточно эффективным инструментом коммуникации по следующим причинам:

1. взаимодействие в определенной степени носит личный характер, так как организация может обращаться непосредственно к конкретному пользователю;
2. мобильное приложение предоставляет широкие возможности для взаимодействия, как между самими пользователями, так и между организацией и внешней средой.

Также пользователи принимают непосредственное участие в развитии мобильного приложения через комментарии (в которых пользователи указывают то, что их устраивает или не устаивает).

Благодаря формам обратной связи организации могут совершенствовать мобильный продукт в соответствии с запросами пользователей. Использование мобильных приложений организацией и постоянное повышение его эффективности как инструмента коммуникации поможет ей повысить свою конкурентоспособность, привлечь новых клиентов и сохранить лояльность старых (повысить лояльность имеющихся), а также сократить многочисленные издержки.

Однако сфера мобильных приложений остается достаточно новой для большинства организаций. Спросу на мобильные приложения пока препятствуют недостаточное количество специалистов в области мобильных разработок, высокая стоимость разработки, поддержки и развития мобильного приложения. В особенности это касается мобильных приложений, ориентированных на внешние коммуникации, которые необходимо постоянно поддерживать и развивать.

Для того чтобы оценить мобильные приложения как инструмент коммуникаций и определить, как можно было бы повысить его эффективность, был проведен анализ отзывов к мобильным приложения «Госуслуги», «Сбербанк Онлайн» и «Наш Санкт-Петербург», находящихся в интернет-магазине приложений Google Play.

Функционал мобильного приложения «Госуслуги» позволяет проверять и оплачивать штрафы, налоговую и судебную задолженности, проверять состояние счетов, подавать заявления, записываться на прием и другое. Однако в процессе использования приложения наблюдается ряд минусов.

Недостатки, выявленные при анализе комментариев:

– сложная процедура оплаты в приложении; неудобство использования некоторых функций в приложении; недоступность некоторых функций в определенных местах (в регионах); отсутствие возможности произвести оплату в приложении по всем или некоторым опциям;
– неполнота и несвоевременность уведомлений; обязательное требование ввода информации личного характера;
– отсутствие указаний на то, где можно найти требуемую к вводу информацию;
– дезинформация пользователя; долгое ожидание внесения изменений в отображаемую информацию;
– отсутствие возможности оплаты частями; не все виды госуслуг представлены в приложении; долгое ожидание ответа от техподдержки.

Также пользователи иногда сталкиваются с техническими проблемами при использовании приложения, например, такими как наличие неработающих функций в приложении, долгая загрузка полного функционала приложения, проблемы с восстановлением пароля, входом в приложение, вводом данных, обработкой и сохранением информации, а также со скоростью работы приложения.

При этом положительные отзывы относились к возможности произведения оплаты, подачи заявлений, получения скидок, просмотра истории события, обращения в техническую поддержку, входа по отпечатку пальца, а также экономии времени на получение услуг.

Мобильное приложение «Наш Санкт-Петербург» дает пользователям возможность отправлять Правительству Санкт-Петербурга сообщения о городских проблемах и тем самым принимать непосредственное участие в жизни города. К описанию проблемы можно добавить фотографию, а также отметить ее на карте города.

Недостатки приложения отмечены следующие:

– долгое ожидание рассмотрения обращений;
– технические сбои в работе приложения;
– неполнота отображаемой информации;
– отсутствие информированности пользователей о существовании подобного приложения;

– трудности с решением проблем через приложение; отсутствие возможности поиска, по ключевым словам, и фамилиям и другие.

Мобильное приложение «Сбербанк Онлайн» имеет также ряд достоинств и недостатков.

Положительные стороны приложения заключаются в том, что оно позволяет оплачивать мобильную связь, ЖКХ, интернет, штрафы, а также осуществлять переводы денежных средств, проверять состояние счетов, находить ближайшие офисы и банкоматы.

Основной недостаток, как замечают пользователи, появился в связи с последним обновлением, призванным усилить безопасность платежей, однако пока повлиял лишь на скорость загрузки в отрицательную сторону. Также с обновлением увеличился вес приложения, затруднилась функция сохранения ранее введенной информации.

Несмотря на то, что мобильные приложения являются достаточно новым инструментом общения, они становятся популярными видами коммуникации. Однако большое количество негативных отзывов о приложениях говорят о том, что их необходимо постоянно развивать и совершенствовать.

На основе исследования отзывов трех популярных мобильных приложений, в которых взаимодействуют компания и потребитель, предложены следующие рекомендации, направленные на повышение эффективности мобильных приложений как инструмента коммуникации:

1. пользователям надо предоставлять возможность выбирать функции при установке приложения, а также ранжировать их в приложении от функций, используемых чаще других к функциям, которые менее используются;
2. у пользователей должна быть возможность самим настраивать представление отображаемой информации: менять цвет, фон, тему, а также масштаб шрифта. Помимо этого, при желании пользователь должен иметь возможность пройти обучение различным функциям приложения;
3. для удобства и повышения скорости использования приложения лучше всего снижать необходимость в выполнении одних и тех же действий.

Кроме вышеперечисленного, следует добавить, что очень часто о существовании многих мобильных приложений пользователи не знают, и получают информацию о новых приложениях через друзей или знакомых. Реже это происходит через рейтинги в магазинах приложений, в которых трудно увидеть все многообразие приложений. Это говорит о том, что организация должна сама заниматься продвижением и распространением своего приложения [2–6].

Резюмируя вышесказанное, можно отметить, что мобильные приложения помогают компаниям устанавливать личную связь с потребителями в определенной степени, увеличивать долю рынка, повышать качество оказываемых услуг и повышать лояльность клиентов, а также в целом оптимизировать и автоматизировать различные бизнес-процессы.

Литература

1. Панфилова А.П. Деловая коммуникация в профессиональной деятельности. – СПб.: Знание, ИВЭСЭП, 2004. – 495 с.
2. Мурет Ч. Империя приложений. – СПб.: Альпина Паблишер, 2013. – 22 с.
3. Приложение «Сбербанк Онлайн» [Электронный ресурс]. – Режим доступа: https://online.sberbank.ru/CSAFront/index.do (дата обращения: 02.12.2018).
4. Приложение «Госуслуги» [Электронный ресурс]. – Режим доступа: https://www.gosuslugi.ru/ (дата обращения: 03.12.2018).
5. Портал «Наш Санкт-Петербург» [Электронный ресурс]. – Режим доступа: http://gorod.kis.gov.spb.ru/ (дата обращения: 03.12.2018).
6. Бороздина Г.В. Психология делового общения: учебник. – Изд. 2-е. – М.: Инфра-М, 2007. – 295 с.

Прижимова Галина Олеговна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P41602
Направление подготовки: 09.04.02 – Компьютерная графика и веб-дизайн
e-mail: g_prij@mail.ru

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004.032

ИССЛЕДОВАНИЕ МЕТОДОВ СОЗДАНИЯ ФУНКЦИОНАЛЬНОЙ WEB-АНИМАЦИИ НА ПЛАТФОРМЕ CSS

Прижимова Г.О.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

Разработка качественного анимационного контента и его уместное применение способно повысить успех web-ресурса в Сети. В условиях высокой конкуренции разработчикам сайтов приходится все большее внимание уделять способам демонстрации информации для потенциальной аудитории. Наглядность и эмоциональное воздействие определили преимущество использования анимационной информации, нежели текстовой или графической. В работе рассмотрены основные подходы для создания функциональной анимации.
Ключевые слова: функциональная веб-анимация, CSS, transition, animation, свойства анимированного объекта, скорость анимации, длительность анимации, переход между состоянием объекта.

Повсеместное внедрение web-сервисов во все сферы человеческой жизни ведет к росту числа сайтов, что является причиной роста конкуренции. Для привлечения целевых клиентов на сайт используются инструменты интернет-маркетинга, которые будут эффективны при условии высокого качества web-ресурса. Ключевыми факторами, определяющими качество сайта, являются: производительность, функциональность, юзабилити и дизайн. Дизайн призван решать не столько проблемы художественного проектирования, сколько проблемы удобства использования конечного продукта потенциальными пользователями. Комфортное взаимодействие пользователя с виртуальной средой достигается применением различных методов к организации и отображению предоставляемой информации. Создание анимационного контента для web-сервисов сегодня широко применяется как один из способов наиболее понятного и емкого предоставления информации. В связи с этим последние несколько лет активно исследуются проблемы, связанные с использованием анимации в веб-дизайне для решения определенного круга задач, а также проблемы выбора методов их решения [1–4].

Анимация интерфейса сегодня предназначена в первую очередь для удобства пользователя. Появилось понятие функциональной анимации – незначительные эффекты с элементами интерфейса, которые подчинены логичной цели. Такая анимация помогает пользователю воспринимать новую информацию.

Преимущество анимации перед другими методами представления контента на сайте объясняется такой функцией человеческого сознания, как кратковременная память. Изменение в интерфейсе, например, скрытие блока информации на сайте, позволяет продлить в сознании пользователя факт изменения, зафиксировать его в памяти, придать новые смыслы.

В анимации интерфейса следует избегать перенасыщенности эффектами. Не стоит использовать анимацию по причине того, что есть техническая возможность. Когда одновременно несколько объектов движутся по экрану в разных направлениях по пересекающимся траекториям, глаз не может охватить все изменения, и отсюда у пользователя возникает чувство растерянности. Переходы необходимо реализовывать четко и просто. При создании перехода важно руководствоваться практической пользой, которую переход даст пользователю.

Также не стоит использовать анимацию, вызывающую большую нагрузку на систему. Хуже, чем неуместная анимация, может быть только анимация, которая замедляет работу сайта.

Функциональная анимация призвана объяснять пользователю происходящее на экране, вести его по изменяющемуся навигационному контексту и поддерживать иерархию составляющих интерфейса.

Дизайн определяет взаимодействие пользователя с интерфейсом, поэтому он должен быть отзывчивым на изменения, иметь обратную связь. Важно, чтобы отклик на действие отражался в месте, где пользователь его осуществил. Связь между действием и последующими за ним изменениями на экране должна быть очевидна.

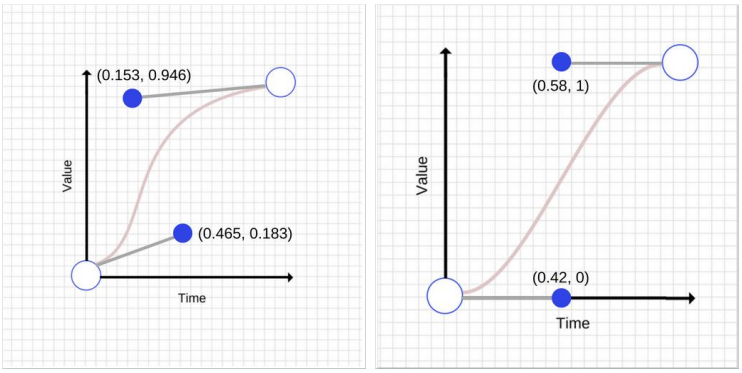
Использование динамики в анимации позволяет повысить качество пользовательского опыта, так как естественные перемещения объектов создают ощущения комфорта при работе с приложениями.

При грамотной настройке таких характеристик анимации как длительность, скорость и плавность создается естественность восприятия пользователем анимации, в некоторых случаях он ее даже может не заметить.

При линейной анимации состояние объекта за одинаковые промежутки времени изменяется равномерно. Такая анимация визуально заметна и может отвлекать.

Анимации, реализованные на CSS или JavaScript дают возможность разрабатывать произвольные функции плавности при помощи кривых Безье. Так разработчик имеет больший контроль над желаемым воздействием, которое должно передаваться пользователю средствами анимации.

Кривая Безье строится по двум парам чисел, которые описывают координаты X и Y опорной точки кубической кривой Безье. Даже незначительное изменение этих значений приводит к заметному изменению кривой, что видно на рисунке.



а б

Рисунок. Кривые Безье со схожими координатами

Использование анимации помогает реализовать плавный переход между различными состояниями объектов на экране и обладает рядом преимуществ перед другими формами

представления информации по донесению информации до целевой аудитории. Однако анимацию все-таки следует относить к второстепенным инструментам дизайна. Ее отсутствие никак не должно оказывать воздействие на работоспособность ресурса или считываемость контента сайта.

Литература

1. Беляев А.А. Анимация в дизайне интерфейса информационных сайтов [Электронный ресурс]. – Режим доступа: http://www.mediascope.ru/node/1618 (дата обращения: 04.01.2019).
2. Анимация в инфографике. Примеры визуализации [Электронный ресурс]. – Режим доступа: http://infographer.ru/animaciya-v-infografike/ (дата обращения: 04.01.2019).
3. Hudson S., Stasko J.T. Animation Support in a User Interface Toolkit: Flexible, Robust, and Reusable Abstractions // Proceedings of the 6th annual ACM Sym-posium on User Interface Software and Technology. Atlanta. – 1993. – V. 15. – P. 19–57.
4. Функциональная анимация для сайта [Электронный ресурс]. – Режим доступа: https://www.100up.ru/blog/funktsionalnaya-animatsiya-dlya-sayta/ (дата обращения: 04.01.2019).

Ростов Данил Сергеевич
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № S4107
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: daniil.rossy@gmail.com

Готская Ирина Борисовна
Год рождения: 1957
Университет ИТМО, факультет программной инженерии и компьютерной техники, д.педагог.н., профессор
e-mail: iringot@mail.ru

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.02

ИССЛЕДОВАНИЕ МЕТОДОВ ИМПЛЕМЕНТАЦИИ РЕНДЕРИНГА КЛИЕНТСКОГО ИНТЕРФЕЙСА НА СТОРОНЕ СЕРВЕРА

Ростов Д.С., Готская И.Б., Государев И.Б.
Научный руководитель – д.педагог.н., профессор Готская И.Б.

Работа выполнена в рамках темы НИР № 615892 «Исследование и разработка в области информационных технологий».

В работе проведено исследование методов имплементации рендеринга клиентского интерфейса на стороне сервера. Рассмотрены положительные и отрицательные характеристики одностраничных веб-приложений, основные подходы при проектировании интерфейсов с требованием рендеринга на сервере, а также использование практик передачи клиентского кода с сервера в браузер.
Ключевые слова: разработка веб-сайтов, одностраничные веб-приложения, рендеринг на стороне сервера, рендеринг одностраничного веб-приложения.

Современные требования к веб-сайтам растут, что удовлетворяет потребность в постоянном совершенствовании существующих технологий и подходов к разработке веб-интерфейсов [1]. Одним из популярных подходов в последнее время стала практика использования одностраничных веб-приложений. Однако при таком подходе происходит изменение классического сценария браузера по рендерингу веб-страниц. В работе рассмотрены методы, позволяющие использовать классический сценарий рендеринга веб-страниц в совокупности с использованием преимуществ одностраничных веб-приложений.

Одним из основных преимуществ одностраничных веб-приложений является возможность динамической генерации HTML-разметки с помощью языка программирования JavaScript. Это позволяет проектировать и реализовывать насыщенные интерактивными элементами веб-интерфейсы, а также удобно и производительно управлять состоянием этих элементов. Также при использовании подхода одностраничных веб-приложений, все взаимодействие с веб-сайтом происходит на единственной HTML-странице. Это является следствием динамического изменения разметки. В дополнении к этому плюсами одностраничных веб-приложений являются производительность и быстрый переход по внутренним ссылкам веб-сайта.

Несмотря на вышеупомянутые плюсы, подход одностраничных веб-приложений имеет некоторые ограничения. Так как вся HTML-разметка веб-сайта генерируется языком JavaScript, то увеличивается нагрузка на веб-трафик пользователя [2]. Это происходит вследствие необходимости доставлять в браузер пользователя большее количество кода, ведь генерирующий разметку JavaScript занимает гораздо больший объем памяти, чем готовая HTML-разметка. Также сгенерированная разметка с помощью JavaScript является трудночитаемой для поисковых роботов, что негативно сказывается на SEO-оптимизации сайта и на интеграции с другими сервисами, с которыми необходимо делиться контентом веб-сайта [3].

Для решения вышеупомянутых проблем с сохранением всех плюсов одностраничных веб-приложений, существуют различные способы рендеринга одностраничного веб-приложения на стороне сервера с целью дальнейшей передачи готовой разметки в браузер пользователя. Эти подходы называются «server-side rendering».

Основной идеей в различных имплементациях «server-side rendering» является возможность запустить JavaScript на сервере, чтобы он мог сгенерировать полученную разметку, затем передать сгенерированную разметку в браузер пользователю и запустить механизм построение DOM-дерева в браузере, чем обеспечивается возможность рендерить веб-страницы по привычному для веб-браузера сценарию [4]. Таким образом, можно выделить три основных этапа в использовании подхода рендеринга одностраничного веб-приложения на стороне сервера. Первый этап – процесс запуска JavaScript кода на сервере и генерация статичных HTML-страниц. Второй этап – оптимальный способ передачи готовой разметке клиенту. Третий этап – запуск полученного кода в веб-браузере и построение DOM-дерева.

Одними из популярных на сегодняшний день инструментами по разработке одностраничных веб-приложений являются: React, Angular, Vue и Ember. Вне зависимости от того, чем является инструмент: фреймворком или библиотекой, каждый из этих инструментов предоставляет свои способы по имплементации рендеринга HTML-разметки на стороне сервера. Безусловным требованием является наличие Node.js, так как без этой платформы JavaScript не сможет быть запущен. Следующим важным требованием является наличие сервера, который будет осуществлять запуск рендеринга по запросу [3]. Также, должны быть предусмотрены способы сериализации и десериализации будущей разметки как на стороне сервера, так и на стороне клиента.

Каждый инструмент по разработке одностраничных веб-приложений может иметь свое готовое решение. Под готовым решением чаще всего подразумевается определенный уровень абстракции над низкоуровневой реализацией вышеупомянутых требований и этапов рендеринга на стороне сервера. К примеру, для библиотеки React существует готовое решение под названием «Next» [5]. Ее основными функциями и возможностями являются: серверная маршрутизация с поддержкой параметризации в адресной строке, динамическое подключение модулей, оптимизация неиспользуемого кода, а также некоторые другие возможности для удобства разработки. Ей аналогом для библиотеки Vue является «Nuxt», функции и возможности которой практически идентично дублируются из «Next». Также существует готовое решение для Angular, которое называется «Universal» [2].

Тем не менее, несмотря на все преимущества использования подходов серверного рендеринга одностраничных веб-приложений, существуют и недостатки. К ним можно отнести: расход ресурсов сервера, увеличение трафика пользователя в некоторых реализациях, увеличение инфраструктуры и сложности приложения, что неизбежно ведет к удорожанию разработки.

В ходе исследования были рассмотрены методы имплементации рендеринга клиентского интерфейса на стороне сервера, их положительные и отрицательные стороны, а также самые популярные готовые решения по их имплементации.

Литература

1. Азимова Д.Ю., Турсунова Д.Г. Современный подход к разработке веб-ресурсов [Электронный ресурс]. – Режим доступа: https://moluch.ru/archive/125/34299 (дата обращения: 25.12.2018).
2. Server-side rendering (SSR): An intro to Angular Universal [Электронный ресурс]. – Режим доступа: https://angular.io/guide/universal (дата обращения: 10.12.2018).
3. Вигерс К., Битти Д. Разработка требований к программному обеспечению. – 3-е изд., доп. // Пер. с англ. – М.: Русская редакция; СПб.: БХВ-Петербург, 2014. – 736 с.
4. Henderson C. Building scalable web sites. – California: O'Reilly Media, 2008. – 352 p.
5. Documentation of Next.js framework [Электронный ресурс]. – Режим доступа: https://nextjs.org/docs/ (дата обращения: 11.12.2018).

Румянцева Александра Геннадьевна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P41602
Направление подготовки: 09.04.02 – Компьютерная графика и веб-дизайн
e-mail: sanya.alexandra@mail.ru

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004.588

АНАЛИЗ ПРИМЕНЕНИЯ ТЕХНОЛОГИЙ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В МУЗЕЙНОМ ПРОСТРАНСТВЕ

Румянцева А.Г.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

В работе рассмотрено применение технологий дополненной реальности в качестве интерактивного дополнения к традиционным экспозициям музейных пространств с целью увеличения внимания общественности. Для этого выполнен обзор существующих на сегодняшний день технологий, реализованных в различных музейных пространствах.
Ключевые слова: дополненная реальность, мультимедиа, интерактивность, 3D-технологии, видеомэппинг.

В век развития цифровых технологий многие музеи не приспособлены к современным реалиям и используют для передачи информации традиционные методы, однако, высокие технологии стремительно проникают в культурное пространство [1–5].

Сегодня далеко не каждого заинтересует классическая музейная экспозиция, поскольку отмечается повышенный интерес к новой, интерактивной подаче информации.

Интерактивность позволяет:
– оставить яркое впечатление об экспонате или экспозиции, благодаря тому, что человек будет участвовать в процессе;
– показать то, что вживую показать невозможно, экспонаты могут быть утеряны, храниться в фондах или габариты не позволяют показать их в экспозиции;
– представить информацию в наглядной форме для разной аудитории;
– осуществить коммуникацию с посетителем.

Интерактивность может быть достигнута различными путями, например, в США в Музее дизайна Купер-Хьюит предлагают своим посетителям гаджет – черную ручку, с помощью которой можно выбрать образец обоев, и на стене появится проекция интерьера соответствующей эпохи. Кроме того, можно нарисовать цветок на интерактивном столе, и посетителю будет предложена возможность послушать про использования цветочных мотивов в дизайне помещений.

Музей искусств Кливленда реализовал так называемую «Стену коллекций». Насчитывается более 3 500 экспонатов, которые можно увеличить и рассмотреть более

детально. Кроме того, есть возможность составить собственную коллекцию-маршрут по экспозиции.

Шведский музей Средиземноморья в Стокгольме использовал 3D-технологии для египетских мумий. Возможность изучать содержимое саркофага и анатомию мумий по слоям появилась благодаря томографическому сканированию и отображению на интерактивном столе.

В то же время российские музеи не отстают от европейских коллег. В музее истории Екатеринбурга посетителям предоставляется возможность полистать интерактивную книгу, на страницах которой представлена ожившая история города. Данный экспонат предполагает использование технологии 3D-видеомэппинга, системы верхней проекции и компьютерной анимации.

Интерактивные песочницы в Тольятти, Самаре, Северном Приладожье и Ханты-Мансийске наглядно иллюстрируют формирование гор и водоемов на планете.

Исходя из вышеизложенных примеров, можно сделать вывод, что с каждым днем мультимедийные технологии все активнее внедряются для формирования эмоциональных впечатлений у посетителя.

Существуют следующие технологии для интерактивной подачи информации в музее:

1. мультимедийные макеты и макеты с дополненной реальностью;
2. дополненная реальность для презентации экспонатов и навигации по музею;
3. виртуальная реальность в музейном пространстве и музей в виртуальном пространстве;
4. объектный видеомэппинг и проекционные инсталляции;
5. мобильные приложения и интерактивные сувениры;
6. интерактивная стена;
7. виртуальный гид.

Применение дополненной реальности позволило бы решить некоторые задачи музейной среды: возможность создания атмосферы для экспонатов, достижение коммуникации между экспонатами и посетителями, при необходимости быстрое изменение экспозиции. Кроме того, дополненная реальность поможет повысить интерес к музеям, а также позволит облегчить процесс восприятия и усвоения нового материала, за счет целенаправленных действий посетителя для получения информации.

Литература

1. Миловидов С.В. Принципы «дополненной реальности» и интерактивная реконструкция в музеях [Электронный ресурс]. – Режим доступа: http://sias.ru/publications/magazines/kultura/2013-4/sotsialnaya-filosofiya-isotsiologiya/848.html (дата обращения: 04.01.2019).
2. Елесин С. Внедрение технологий виртуальной и дополненной реальности в музейную практику: проблемы и решения // Цифровая гуманитаристика: ресурсы, методы, исследования. – 2017. – С. 174–177.
3. Родионова Д.Д., Сергеев А.В. Технология дополненной реальности как перспективное направление развития музейного пространства на современном этапе // Вестник КемГУКИ. – 2015. – № 33. – С. 51–57.
4. Гниятова Е.В., Ройз Е.Е. Трансформация музейного пространства в условиях новых визуальных стратегий // Изв. Томского политехнического университета. – 2013. – Т. 303. – № 6. – С. 297–300.
5. Приложения дополненной реальности: будущее реально и виртуально [Электронный ресурс]. – Режим доступа: http://arnext.ru/articles/prilozheniya-dopolnennoy-realnosti-budushhee-realno-i-virtualno-7479 (дата обращения: 04.01.2019).

Рунаева Виолетта Владиславовна
Год рождения: 1996
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P41622
Направление подготовки: 09.04.04 – Программная инженерия
e-mail: runaevavioletta@gmail.com

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.946

ОСОБЕННОСТИ ПРИМЕНЕНИЯ ТЕХНОЛОГИИ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ В ВЕБ-БРАУЗЕРЕ

Рунаева В.В., Государев И.Б.
Научные руководители: к.педагог.н., доцент Государев И.Б.;
к.педагог.н., доцент Шуклин Д.А.

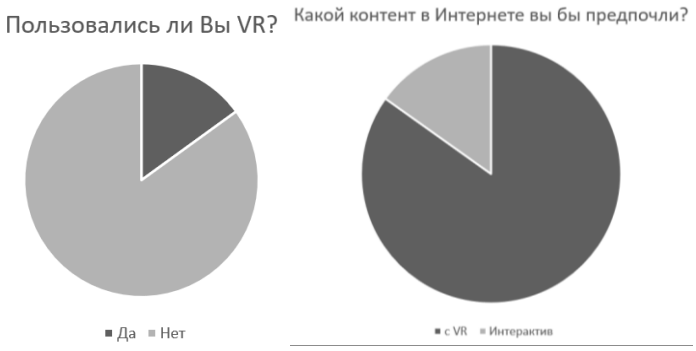
В работе рассмотрены общие представления о понятии и принципах работы виртуальной реальности в браузере. Исследована заинтересованность и вовлеченность пользователей в технологию виртуальной реальности, на основе полученных результатов проанализированы преимущества и недостатки технологии WebVR.
Ключевые слова: виртуальная реальность, браузер, Интернет, WebVR.

Главным трендом в области развития интерактивных технологий на данный момент является виртуальная реальность. Появление технологий виртуальной реальности предоставляет возможность получить новый опыт взаимодействия с контентом и позволяет погрузить пользователя в среду, симулирующую различные визуальные модели. Стремительное распространение технологии виртуальной реальности свидетельствует о тенденции отразить высокую значимость IT-технологий в повседневной жизни человека.

Еще совсем недавно представление о виртуальной реальности (VR) ограничивалось только играми и не представлялось возможным совмещения таких понятий, как «веб-браузер» и «виртуальная реальность». Интерес к VR-технологиям возрос, когда появились устройства для их интеграции с веб-браузерами. Множество крупных компаний, таких как: Google, Facebook, Mozzila, Samsung, вкладывают инвестиции в развитие WebVR. Таким образом, можно отметить стремительный рост спроса и положительную тенденцию на данную технологию.

В ходе изучения данной предметной области было произведено социологическое исследование на предмет заинтересованности и вовлеченности пользователей в технологию VR. Опрашивались пользователи социальных сетей Instagram и ВКонтакте в количестве трехсот человек. По результатам опроса были сделаны следующие выводы, актуальные на 2019 год:

1. только 15% опрашиваемых пробовали использовать VR как представлено на рис. 1, а;



а б

Рис. 1. Результат опроса по использованию VR (а) и по предпочтению контента в Интернете (б)

2. как видно из рис. 1, б, 83% людей выбрали бы веб-ресурс с контентом в формате VR, 17% – интерактивный материал;
3. 80% опрашиваемых не хотят скачивать дополнительное программное обеспечение (ПО) для использования VR, как проиллюстрировано на рис. 2, а;

а б

Рис. 2. Результат опроса по способу работы с VR (а) и по доступности VR-устройств (б)

4. 54% опрашиваемых могут себе позволить купить VR-очки, как показано на рис. 2, б.

В процессе опроса выявлены основные причины отсутствия высокого спроса на технологию VR:

1. недостаточное просвещение людей о данной технологии;
2. малое количество доступных вариантов использования VR, помимо игровой сферы;
3. высокая стоимость на устройства для работы с VR;
4. негативное влияние на физическое и психологическое состояние человека.

На рынке IT-технологий реализация объекта в пространстве VR становится все более востребована и применима ко многим сферам жизни человека, а VR-устройства становятся все доступнее потребителям с каждым годом. Повысить спрос на использование VR может применение данной технологии в браузере. На данный момент WebVR – малоизведанная территория в мире IT-технологий и только развивается, в связи с этим возникают вопросы, связанные с понятием, применением и способом создания такого приложения.

Веб-виртуальная реальность (WebVR) – это открытая спецификация, которая позволяет использовать технологию VR в веб-браузере и представляет собой экспериментальный программный интерфейс программирования (API) на языке JavaScript. Цель WebVR заключается в том, чтобы предоставить пользователю возможность получить опыт взаимодействия с виртуальной реальностью, независимо от того, какое устройство он имеет. Понятие WebVR существует с 2014 года и предложено компанией Mozilla, однако первую спецификацию по технологии компания выпустила только в 2016 году. Технология WebVR проста в восприятии, так как она плавно работает на большинстве веб-браузерах смартфонов [1].

На данный момент HTML5 – язык разметки гипертекста – является текущим стандартом для создания домашних страниц. Однако WebVR поддерживает распространение трехмерных

моделей через Интернет, которые имеют все характеристики виртуальных моделей. Они основаны на полигональном представлении и имеют возможность отображаться в анимированном виде и включать в себя функциональность и динамическое поведение, а также могут интерактивно контролироваться пользователем. Уже сейчас страницы могут быть заменены «домашними пространствами» с гиперссылками на другие пространства или страницы, расположенные на любом сервере во всемирной сети Интернет.

WebVR API работает по следующему принципу, цикл работы WebVR представлен на рис. 3 [2]:

1. запрашивает список доступных VR-устройств;
2. выполняет проверку – поддерживает ли необходимое устройство режимы представления, которые необходимы веб-приложению. Если проверка прошла успешно, то приложение объявляет VR-функциональность пользователю;
3. запускает сеанс VR для предоставления контента и цикл рендеринга, которые создают графические кадры для отображения на VR-устройстве;
4. прекращает цикл рендеринга и завершает сеанс, когда пользователь указывает на желание выйти из системы.

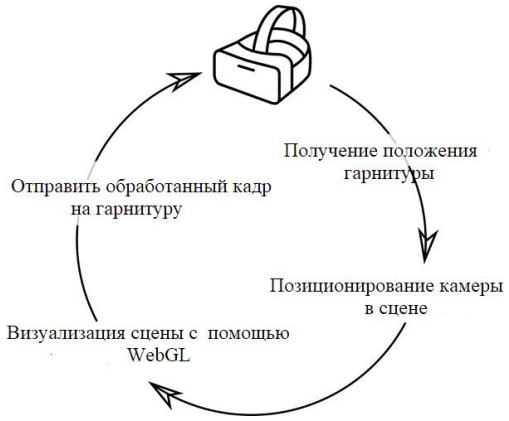


Рис. 3. Цикл работы WebVR

Две основные платформы VR являются полными противоположностями: высокопроизводительные персональные компьютеры с усовершенствованными контроллерами и среднего/высокого уровня мобильные телефоны, имеющие либо один единственный контроллер, либо вообще ни одного. В связи с этим возникают проблемы:

1. поддержка постоянной скорости передачи фреймов (кадров) на платформах с разными возможностями;
2. обеспечение хорошего пользовательского опыта VR на устройствах с различными параметрами.

Основная цель стратегии прогрессивного улучшения – сделать доступным контент и функциональные возможности веб-страниц любого браузера.

Технология WebVR имеет следующие преимущества:

– для работы с WebVR-приложением не требуется устанавливать дополнительное ПО,
– работа ведется напрямую из веб-браузера;
– встроенное кэширование веб-браузера позволяет избавиться от лишней занимаемой памяти;
– для быстрого распространения контента VR в сети Интернет имеется возможность делиться URL-адресом веб-ресурса;
– сеть Интернет обладает высокоуровневыми интерфейсами для API более низкого уровня, что обеспечивает современные технологии для каждого разработчика;
– выполнение кода в сети Интернет наиболее производительно по сравнению с ПО, за счет использования на стадии разработки JavaScript API, которые могут быть легко оптимизированы и предварительно скомпилированы;

– интерпретация языка программирования в соответствии с используемой платформой достигается за счет применения декларативного языка HTML5.

Согласно отчету по тенденциям использования VR, представленному на рис. 4, от платформы по статистике Sketchfab, бурное развитие технологий VR коснулось практически всех сфер деятельности человека. Однако не все области применения VR рационально использовать в веб-браузере, так, например, использование VR в промышленности или при расчетах в конструировании имеет смысл использовать только на производствах в качестве нативных приложений.

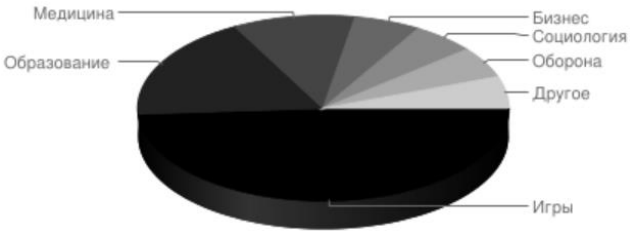


Рис. 4. Отчет по тенденциям VR от Sketchfab

VR в веб-браузере целесообразно использовать для:

– получения новых развития социальных навыков. Так, например, приложение SkillHub позволяет тренировать публичное выступление, интерактивно моделировать диалог, тренировать эмоциональный интеллект;
– терапевтических целей, в качестве тренажера-симулятора для избавления от фобий и параной;
– трансляции спортивных матчей, концертов, комедийных шоу – DOM2AFRAME [3];
– визуализации новостного контента. Так, например, в своих репортажах виртуальное погружение в новости применяют такие компании, как ABC и Frontline [4];
– изучения искусства и культуры – посещение музейных залов и выставок, просмотр исторических панорамных фильмов;
– визуализации гостиничных номеров и предоставляемые туры – The Suit Life;
– визуализации продаваемой недвижимости – Sotheby’s International Reality;
– исследования планеты – Goolge EarthVR [5].

По результатам исследования сделан вывод, что технология VR актуальна и востребована на сегодняшний день, однако, слабо развита в сферах, отличных от игровой. Таким образом, WebVR представляет собой новый способ восприятия информации в сети Интернет. Такой подход упрощает получение доступа к VR-приложению для пользователя и увеличивает охват аудитории.

Литература

1. Ye Q. VR Interactive Feature of HTML5-based WebVR Control Laboratory by Using Head-mounted Display // International Journal of Online Engineering. – 2018. – V. 14. – № 03. – P. 20–33.
2. Hadjar H. WebVR based Interactive Visualization of Open Health Data // Proceedings of the 2nd International Conference on Web Studies. – 2018. – P. 56–63.
3. Marx R. DOM2AFRAME: Putting the web back in WebVR // 3D Immersion (IC3D), 2017 International Conference on. – 2017. – P. 1–8.
4. Ebola outbreak a virtual journey [Электронный ресурс]. – Режим доступа: https://www.pbs.org/wgbh/frontline/announcement/ebola-outbreak-a-virtual-journey-frontlines-first-virtual-reality-documentary-debuts-at-ona-on-sept-26-press-release/ (дата обращения: 04.01.2019).
5. Google Earth VR [Электронный ресурс]. – Режим доступа: http://vr.google.com/earth/ (дата обращения: 04.01.2019).

Сенин Артем Дмитриевич
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4263
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: artemee@corp.ifmo.ru

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.021

ПРИМЕНЕНИЕ РЕАКТИВНОЙ ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ В УПРАВЛЕНИИ СОСТОЯНИЕМ ВЕБ-ПРИЛОЖЕНИЙ

Сенин А.Д., Государев И.Б.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

В работе исследован вопрос применения модели реактивного программирования в веб. Рассмотрены основные составляющие и примеры использования модели реактивного программирования в вебе, для хранения и работы с состоянием одностраничных веб-приложений.
Ключевые слова: модели программирования, реактивное программирование, SPA, паттерны проектирования, веб-приложения, технологии, информация.

Развитие веб-технологий и пользовательских интерфейсов актуализировало проблему работы с большими объемами данных, представляемых в интерфейсе, скорости обновления пользовательского интерфейса в ответ на действия пользователя, хранения состояния веб-приложений.

В современных интерфейсах веб-приложений большое внимание уделяется хранению и обработке данных, которые они представляют. Данные могут быть получены как со стороны сервера, так и в процессе работы приложения, например, от пользователя [1]. Соответственно в процессе навигации в приложении, данные, которые в данный момент не используются, нельзя потерять и необходимо где-то сохранить. Объект, отвечающий за работу с данными приложения, называется его состоянием.

Одни и те же данные, хранимые в состоянии, могут быть отражены в различных частях приложения, в том числе и одновременно. Отсюда возникает вопрос синхронизации и их обновления в интерфейсе в случае изменения [2]. Изменения могут быть асинхронными, например, обновленные данные могут приходить со стороны сервера.

В таком случае необходима некоторая модель, которая бы описывала хранимые данные в их текущем состоянии. Такой моделью является реактивная модель программирования [3]. В ее основе паттерн «наблюдатель». Этот паттерн реализуется при помощи двух объектов, один из них называется Observable и отвечает за добавление, удаление, оповещение наблюдателей, и второй – Observer – наблюдатель, который подписывается на изменения и выполняет необходимые действия для их обработки [1].

Шаблон «наблюдатель» применяется в системе, когда она имеет объект, рассылающий сообщения, и имеет одного и более получателей сообщений. При этом отправитель не обязан знать, что именно делают с его сообщениями наблюдатели.

Хранение состояния веб-приложения связано с обработкой данных и отображением их в некотором представлении. Таким образом, приложение содержит две основные части, которые находятся в фокусе внимания данной работы: данные и представления [4].

Разные представления могут оперировать одними и теми же данными. В случае изменения данных необходим механизм для обновления всех связанных с этими данными представлений.

За работу с данными отвечают модели, которые предоставляют данные представлениям. Взаимодействие моделей и представлений показано на рис. 1, а. Как следует из иллюстрации, связь между моделями и представлениями может быть достаточно сложной, что вызывает сложность в обновлении данных при одновременной работе двух представлений с одной моделью.

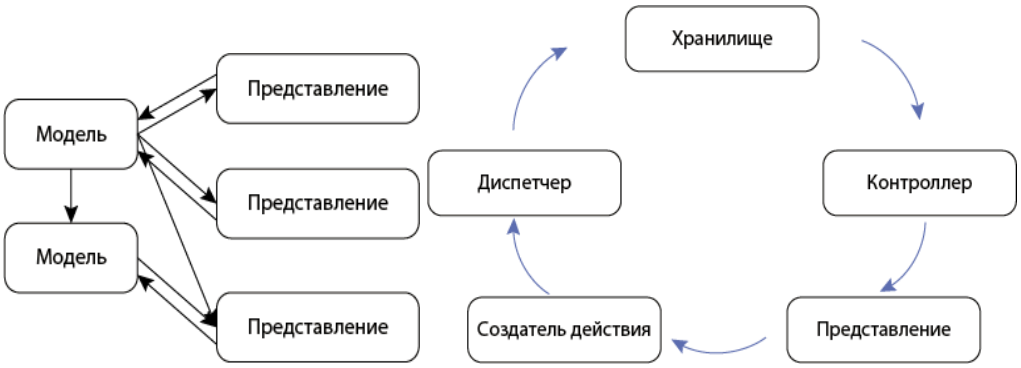


Рис. 1. Взаимодействие моделей и представлений (а); участников Flux-архитектуры (б)

В качестве одного из подходов к решению проблемы взаимодействия моделей и представления следует выделить архитектуру Flux [4]. При таком подходе данные передаются только в одном направлении, сверху вниз, т.е. от родительских компонентов к дочерним.

За хранение данных отвечает единственный источник истины – хранилище. Оно является главным объектом, отвечающим за работу с данными. Изменение и обновление данных осуществляется при помощи редьюсеров, являющихся частью хранилища и отвечающих за изменение определенных частей хранилища, разделенных по типу данных.

Таким образом, вся логика изменения и обновления данных сосредоточена в хранилище, что гарантирует консистентность данных и последовательное, а не одновременное их изменение.

В архитектуре Flux хранилище является отправителем сообщений (Observable), а компоненты являются получателями сообщений (Observers) [5, 6]. Визуально данная архитектура представлена на рис. 1, б.

Компоненты (представления) в этом случае вызывают изменение состояния при помощи действий (actions), которые указывают тип действия и данные, которые необходимо внести в хранилище, или указание на данные, которые необходимо из хранилища удалить.

Рассмотрим общий принцип работы Flux-архитектуры, а также как именно в ней применяется реактивный подход к программированию.

В качестве примера будет рассмотрен счетчик, хранящий число, которое можно увеличить на единицу, либо уменьшить при помощи нажатия соответствующей кнопки в интерфейсе. Данные данного счетчика будут отображаться в двух компонентах: заголовке и текстовом блоке.

Во-первых, необходимо создать редьюсер, который будет позже передан хранилищу для работы. Данный редьюсер принимает на вход тип действия, которое необходимо выполнить. Редьюсер является чистой функцией, т.е. не изменяет существующее состояние, а возвращает его измененную копию, таким образом, при одинаковых входных данных редьюсер будет всегда возвращать одинаковый результат.

```
const counter = (state = 0, action) => {
```

```
  switch (action.type) {
        case 'INCREMENT’:
              return state + 1;
        case 'DECREMENT’:
              return state - 1;
        default:
              return state;
  }
}
```

После того, как создан редьюсер, он передается хранилищу, которое содержит состояние, список подписчиков, метод, возвращающий текущее состояние.

```
const createStore = (reducer) => {
  let state;
  let listeners = [];
  const getState = () => state;
}
```

При создании хранилища используется реактивная модель для подписки на изменения. В качестве ответа метод подписки возвращает функцию отмены подписки на изменения.

```
const createStore = (reducer) => {
  …
  const subscribe = (listener) => {
        listener();
        listeners.push(listener);
        return () => {
              listeners = listeners.filter(
                    l => l !== listener
              );
        }
  }
  …
}
```

Для вызова действия представлениям будет представлен метод dispatch, являющийся фабрикой actions. Здесь используется реактивная модель для рассылки сообщений подписчикам.

```
const createStore = (reducer) => {
  …
  const dispatch = (action) => {
        state = reducer(state, action);
  listeners.forEach((listener) => listener());
  }
  dispatch({});
  …
}
```

В момент запуска приложения подписчиками являются методы, отображающие в соответствующие теги текущее состояние счетчика.

Запуск функции, создающей хранилище. Слушатели событий – функции, использующие текущее состояние.

```
const store = createStore(counter);
const heading = () => { document.querySelector('#heading span').innerHTML = store.getState(); }
const text = () => { document.querySelector('#text span').innerHTML = store.getState(); }
```

Подписка функций на изменения состояния.
const headingListener = store.subscribe(heading);
const textListener = store.subscribe(text);
Использование функции отказа от подписки.
document.querySelector('#heading button').addEventListener('click', headingListener);
document.querySelector('#text button').addEventListener('click', textListener);

Результат запуска приложения показан на рис. 2. Первоначально данные обновляются в заголовке и блоке текста. После чего, используя функцию отписки, оба элемента перестают получать обновления, в то время как счетчик обновляется. Если выполнить подписку блока текста, то его содержимое автоматически будет обновлено, и при изменении состояния счетчика будет меняться и значение в тестовом блоке. Если же выполнить функцию отписки, сохраненную в переменной *a*, то блок текста снова перестанет получать обновляемые данные состояния приложения.

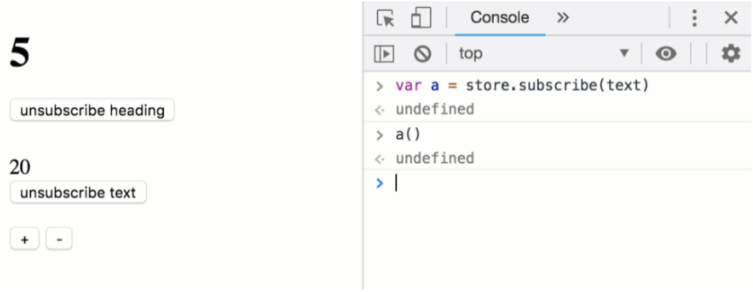


Рис. 2. Запуск приложения

Таким образом, была выявлена связь реактивной модели программирования и обработки состояния веб-приложений. Реактивная модель программирования позволяет эффективно управлять как состоянием веб-приложения, так и отображением этого состояния в связанных, использующих эти данные, представлениях [1, 2, 4].

Реактивное программирование, будучи примененным в сфере веб, решает очень важные задачи, связанные с реализацией как пользовательских интерфейсов, так и работы с данными, представляющими состояние веб-приложения, обеспечивая удобную с точки зрения написания кода программы и последующей поддержки продукта [7].

Литература

1. Evan C. Asynchronous functional reactive programming for GUIs // Programming Language Design and Implementation. – 2013. – P. 411–422.
2. Маркин Е.И., Рябова К.М. Использование реактивного программирования при разработке мобильных приложений // Computational nanotechnology. – 2016. – № 2. – С. 170–173.
3. Манифест реактивного программирования: портал [Электронный ресурс]. – Режим доступа: http://www.reactivemanifesto.org/ (дата обращения: 11.01.2019).
4. Mezzalira L. Front-End Reactive Architectures. – Apress, Berkeley, CA, 2018. – 188 p.
5. Маркин Е.И., Рябова К.М. Реактивное программирование // Вестник современных исследований. – 2016. – № 2-1(2). – С. 112–117.
6. Паттерн Observer (наблюдатель, издатель-подписчик) [Электронный ресурс]. – Режим доступа: http://cpp-reference.ru/patterns/behavioral-patterns/observer/ (дата обращения: 16.01.2019).
7. Tom V.C. A Survey on Reactive Programming [Электронный ресурс]. – Режим доступа: https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwiT9LCPhJLkAhUvtYsKHSwLCLgQFjAAegQIAhAC&url=http%3A%2F%2Fsoft.vub.ac.be%2FPublications%2F2012%2Fvub-soft-tr-12-13.pdf&usg=AOvVaw1Leouxp7ZT0tOksVrmi8_I (дата обращения: 04.01.2019).

Сергиенко Андрей Александрович
Год рождения: 1994
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4270
Направление подготовки: 09.04.02 – Мультимедиа-технологии
в искусстве театра, кино и телевидения
e-mail: master-ac@yandex.ru

Рущенко Нина Геннадиевна
Год рождения: 1972
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: rushchenko@corp.ifmo.ru

Сизиков Валерий Сергеевич
Год рождения: 1939
Университет ИТМО, факультет программной инженерии
и компьютерной техники, д.т.н., профессор
e-mail: sizikov2000@mail.ru

УДК 004.932, 519.642.3

НОВЫЙ БЫСТРЫЙ АЛГОРИТМ ВОССТАНОВЛЕНИЯ СМАЗАННЫХ ИЗОБРАЖЕНИЙ

Сергиенко А.А., Рущенко Н.Г.
Научный руководитель – д.т.н., профессор Сизиков В.С.

Работа выполнена в рамках темы НИР № 619296 «Разработка методов создания и внедрения киберфизических систем».

В данной работе изложен новый быстрый алгоритм восстановления смазанных изображений путем решения некорректного двумерного интегрального уравнения Фредгольма I рода типа свертки методом параметрической фильтрации Винера и для сравнения методом регуляризации Тихонова с использованием преобразования Фурье. В алгоритме повышение скорости решения интегрального уравнения типа свертки достигается за счет использования быстрого преобразования Фурье и предварительного вычисления ряда спектральных функций, не зависящих от смазанного изображения (правой части интегрального уравнения) g. Данный алгоритм может быть применен для восстановления в реальном времени изображений w по смазанным изображениям g подвижных объектов (самолетов, автомобилей и т.д.). Приведен численный пример быстрого (в течение около 0,1 с) восстановления смазанного под углом изображения самолета.
Ключевые слова: смазывание изображения, функция рассеяния точки, интегральное уравнение, метод параметрической фильтрации Винера, «заготовленные» функции, MATLAB.

Введение. Изображения объектов (людей, самолетов, автомобилей, технических деталей, текстов, космических объектов, наблюдаемых в телескоп, земной поверхности из ближнего космоса, вирусов в микроскопе и т.д.) могут быть искажены – смазаны,

дефокусированы, зашумлены [1–3]. В данной работе рассмотрено равномерное и прямолинейное смазывание изображений, вообще говоря, под углом. Искажения изображений могут быть устранены путем их математической обработки. Часто требуется выполнить восстановление изображений в реальном времени. Данная работа является продолжением работ [4–6].

Причинами смаза изображений объектов могут быть: смещение устройства регистрации изображений (фотоаппарата, видеокамеры и т.д.), рассогласование движения конвейера с техническими деталями и камеры параллельного слежения, отличие угловых скоростей вращения небесной сферы и телескопа, а также движение самого объекта (самолета, автомобиля, бегуна на дорожке и т.д.) за время экспозиции [4–6].

В работах [4–6] рассмотрен ряд быстрых алгоритмов восстановления смазанных изображений, причем получены оценки количества необходимых операций и ячеек памяти компьютера. При этом в этих работах рассмотрено быстрое восстановление изображения путем решения набора одномерных интегральных уравнений с направлением оси x вдоль смаза, а оси y – перпендикулярно смазу (приведены примеры лишь с горизонтальным смазом). Однако такой подход неудобен тем, что нужно подстраивать оси x и y под смаз. В работе [6] рассмотрено также быстрое восстановление изображения путем решения одного двумерного интегрального уравнения (ИУ), причем ось x при любом направлении смаза направлена горизонтально, а y – вертикально. Во всех работах [4–6] для восстановления изображений путем решения интегральных уравнений использован метод регуляризации Тихонова (РТ).

В данной работе продолжено рассмотрение быстрого восстановления смазанных изображений путем решения двумерного ИУ методом параметрической фильтрации Винера (ПФВ) и выполнено его сравнение с методом РТ.

Математическая формулировка задачи устранения смаза изображения. Задача устранения смаза (а также дефокусирования) изображения может быть выполнена путем решения двумерного ИУФредгольма I рода типа свертки [2, 3, 6]:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x-\xi, y-\eta)\, w(\xi,\eta)\, d\xi\, d\eta = g(x,y) + \delta g\,, \qquad (1)$$

где h – функция рассеяния точки (ФРТ, PSF) или ядро ИУ, как правило, пространственно-инвариантная (разностная) функция; w и g – распределение интенсивности по истинному и искаженному изображениям соответственно; δg – шум; ось x (и ξ) направлена горизонтально слева направо, а ось y (и η) – вертикально вниз и оси не связаны с направлением смаза изображения (лишь построение ФРТ h зависит от направления смаза, см. дальше рис. 2).

Задача решения ИУ (1) является некорректной (существенно неустойчивой) [2–6]. В работах [4–6] для его устойчивого решения использован метод РТ.

Решение ИУ (1) методом РТ с использованием преобразования Фурье (ПФ) равно $w_\alpha(x,y) = F^{-1}\left(W_\alpha(\omega_1,\omega_2)\right)$, где F^{-1} – обратное ПФ; $\alpha>0$ – параметр регуляризации; $W_\alpha(\omega_1,\omega_2)$ – регуляризованный спектр решения (ω_1, ω_2 – часто́ты Фурье) [3]:

$$W_\alpha(\omega_1,\omega_2) = \frac{H^*(\omega_1,\omega_2)}{\left|H(\omega_1,\omega_2)\right|^2 + \alpha\,(\omega_1^2+\omega_2^2)^p}\, G(\omega_1,\omega_2)\,, \qquad (2)$$

где $p \ge 0$ – порядок регулязации (обычно p=1, 2 или 3), а $H(\omega_1,\omega_2) = F(h(x,y))$ и $G(\omega_1,\omega_2) = F(g(x,y))$ – спектры (ПФ), причем F – прямое ПФ.

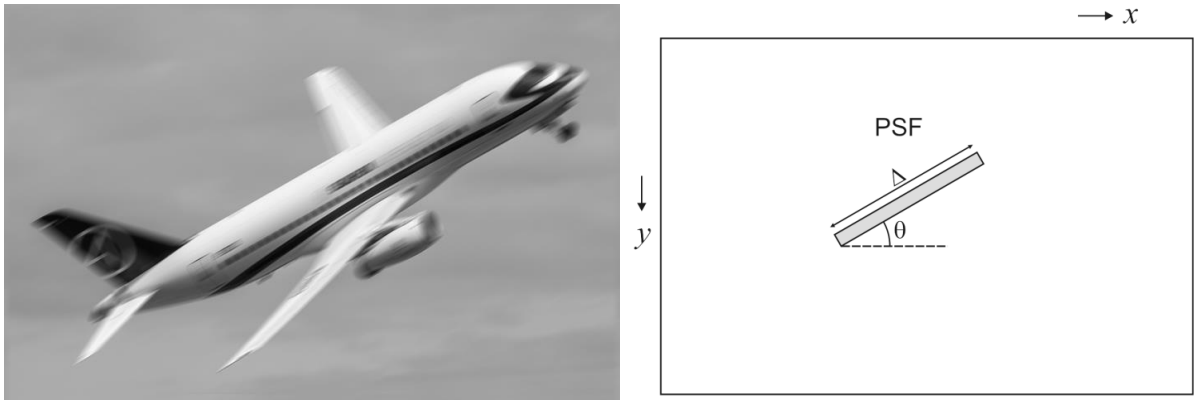
Используем также метод ПФВ [2, 3], согласно которому решение ИУ (1) равно $w_K(x,y) = F^{-1}\left(W_K(\omega_1,\omega_2)\right)$, где $K \ge 0$ – параметр метода (отношение шум/сигнал по мощности (NSPR)), а отфильтрованный спектр решения равен

$$W_K(\omega_1,\omega_2)=\frac{H^*(\omega_1,\omega_2)}{\left|H(\omega_1,\omega_2)\right|^2+K}G(\omega_1,\omega_2)\,. \qquad (3)$$

Решение ИУ (1) методами РТ и ПФВ реализовано в пакете IPT-системы MATLAB в виде m-функций deconvreg.m и deconvwnr.m соответственно [7].

В данной работе основное внимание уделено методу ПФВ и сравнению его с методом РТ в задаче восстановления смазанных изображений.

В качестве иллюстрации на рис. 1, а, представлено смазанное под углом изображение.



а б

Рис. 1. Смазанное под углом изображение самолета, файл SSJ100.jpg 478×744 пкс (а); ФРТ в виде узкой полоски (б)

В этом случае при дискретизации задачи ФРТ h будет PSF-квазидиагональной матрицей в виде узкой полоски [3, С. 112] (рис. 1, б). Это нашло отражение в IPT-пакете m-функций системы MATLAB, где вычисление $h(x,y)$ запрограммировано в m-функции fspecial.m [7].

Значения параметров смаза: величина смаза Δ и угол смаза θ заранее (до появления самолета) неизвестны точно, а ошибки в значении Δ даже в 1–2 пкс и в значении θ в 1–2 град обычно ведут к значительной погрешности в восстановленном изображении из-за неустойчивости задачи (см. [8]).

Быстрый алгоритм. Воспользуемся методом ПФВ для восстановления изображения, смазанного под углом θ, вообще говоря, ненулевым. Предлагаем следующую схему быстрого восстановления изображения на примере трех значений Δ и трех значений θ, поскольку точные значения Δ и θ заранее неизвестны.

1. Рассчитываем заранее 9 ФРТ $h(x,y)$ для трех значений Δ и трех значений θ с помощью MATLAB-оператора: PSF=fspecial('motion', Delta,theta); получим 9 PSF 11×19 и запоминаем их в памяти компьютера.
2. Рассчитываем заранее 9 спектров двумерных PSF (Optical Fransfer Function, OTF): $H(\omega_1,\omega_2)=F(h(x,y))$ следующим образом: H=psf2otf(PSF,size(g)); или H=fft2(PSF); [7].
3. Заводим сетки узлов по ω_1 (вертикально) и ω_2 (горизонтально). Пусть m и n – число строк и столбцов в матрицах изображений w и g. Шаги дискретизации по x и y равны $\Delta x=\Delta y=1$ пкс. Поэтому часто́ты Найквиста равны $\omega_{1\max}=\pi/\Delta y=\pi$ и $\omega_{2\max}=\pi/\Delta x=\pi$. Шаги дискретизации по ω_1 и ω_2 равны $\Delta\omega_1=\omega_{1\max}/(m-1)$ и $\Delta\omega_2=\omega_{2\max}/(n-1)$. В результате сетки узлов по ω_1 и ω_2 равны $\omega_1=0,\Delta\omega_1,\ldots,\omega_{1\max}$ и $\omega_2=0,\Delta\omega_2,\ldots,\omega_{2\max}$.
4. Выражение (3) запишем в виде:

$$W_K(\omega_1,\omega_2)=M_K(\omega_1,\omega_2)G(\omega_1,\omega_2)\,, \qquad (4)$$

где

$$M_K(\omega_1,\omega_2)=\frac{H^*(\omega_1,\omega_2)}{\left|H(\omega_1,\omega_2)\right|^2+K} \qquad (5)$$

– множитель, который может быть вычислен заранее для нескольких «типичных» значений Δ, θ и *K*. При этом рассчитываем $H^*(\omega_1,\omega_2)$: Hc=conj(H); и $\left|H(\omega_1,\omega_2)\right|^2$: H2=abs(H).^2; Получаем и запоминаем 27 множителей $M_K(\omega_1,\omega_2)$.

5. Когда будет получено смазанное изображение $\tilde{g}(x,y)$, вычисляем спектр (двумерное ПФ от $\tilde{g}(x,y)$) $G(\omega_1,\omega_2)=F(\tilde{g}(x,y))$ с помощью MATLAB-оператора: G=fft2(g).
6. Решение ИУ (1) методом ПФВ и двумерного ПФ (восстановленное изображение) равно $w_K(x,y)=|F^{-1}(W_K(\omega_1,\omega_2))|$, где отфильтрованный спектр решения $W_K(\omega_1,\omega_2)$ выражается формулой (4). Будет получено 27 возможных спектров $W_K(\omega_1,\omega_2)$ и 27 возможных решений $w_K(x,y)$.

Назовем алгоритм (1), (4), (5) двумерным быстрым алгоритмом. Он является быстрым за счет предварительного вычисления тех спектральных функций, которые не зависят от измеренного смазанного изображения *g*. Данный алгоритм отличается от быстрых алгоритмов работ [4, 5] тем, что в нем оси *x* и *y* не привязаны к смазу и, кроме того, используется двумерная обработка (решение одного двумерного ИУ (1)).

Оценка числа операций и памяти. Оценим, сколько требуется операций умножения/деления для реализации двумерного быстрого алгоритма. Основное время будет тратиться на вычисление $G(\omega_1,\omega_2)$ и $w_K(x,y)$, а $H(\omega_1,\omega_2)$ и $M_K(\omega_1,\omega_2)$ вычисляются заранее, поэтому не будем подсчитывать число операций для их вычисления.

Полагаем, что вычисления двумерных быстрых преобразований Фурье (БПФ) $G(\omega_1,\omega_2)$ и обратных БПФ (ОБПФ) $w_K(x,y)$ выполняются путем вычисления одномерных БПФ сначала по каждой из m строк, а потом по каждому из n столбцов [3, С. 71]. В этом случае число операций для вычисления одного $G(\omega_1,\omega_2)$ и одного $w_K(x,y)$ равно $\approx 2mn\log_2 n+2nm\log_2 m=2mn\log_2(mn)$. Если $m=n$, то число операций будет $\approx 4n^2\log_2 n$. Для размещения $G(\omega_1,\omega_2)$, $M_K(\omega_1,\omega_2)$, $W_K(\omega_1,\omega_2)$ и $w_K(x,y)$ в виде дискретных массивов потребуется по mn ячеек, т.е. всего $4mn$ ячеек памяти. Если $m=n$, то потребуется $4n^2$ ячеек компьютерной памяти. При этом для обработки всех 27 вариантов потребуется в 27 раз больше операций и памяти.

Таким образом, двумерный быстрый алгоритм на основе метода параметрической фильтрации Винера требует таких же ресурсов, как и алгоритм на основе метода регуляризации Тихонова. Если, например, $m=n=400$, то для вычисления одного БПФ $G(\omega_1,\omega_2)$ и одного ОБПФ $w_K(x,y)$ потребуется примерно $4n^2\log_2 n=5{,}5$ млн операций. Если скорость компьютера около 1 млрд оп/с, то на восстановление одного изображения потребуется около 0,01 с. А для восстановления всех 27 возможных изображений потребуется около 0,1 с.

Численный пример. Рассмотрим следующий пример. На рис. 1, а, дано смазанное под углом и слабо зашумленное (шум 0,3%) изображение самолета на взлете. Точные значения угла смаза θ и величины смаза Δ неизвестны. Но на практике в условиях реальной работы аэропорта обычно известны для конкретного самолета диапазоны значений угла взлета самолета θ и смаза Δ. Полагаем, что $\theta\approx 29-31°$, $\Delta\approx 19-21$ пкс (диапазоны значений θ и Δ).

Для восстановления изображения используем двумерный быстрый алгоритм (1), (4), (5). Для значений $\theta=29,30,31$; $\Delta=19,20,21$ заранее (до взлета самолета) рассчитываем

(«заготавливаем») 9 ФРТ $h(x, y)$ с помощью m-функции fspecial.m и 9 спектров $H(\omega_1, \omega_2)$ с помощью m-функции psf2otf.m или fft2.m [7].

Далее для трех значений K, например, $K = 10^{-4}$, 10^{-3}, 10^{-2} рассчитываем $M_K(\omega_1, \omega_2)$ согласно (5). Получим 27 «заготовок» $M_K(\omega_1, \omega_2)$ для разных θ, Δ и K.

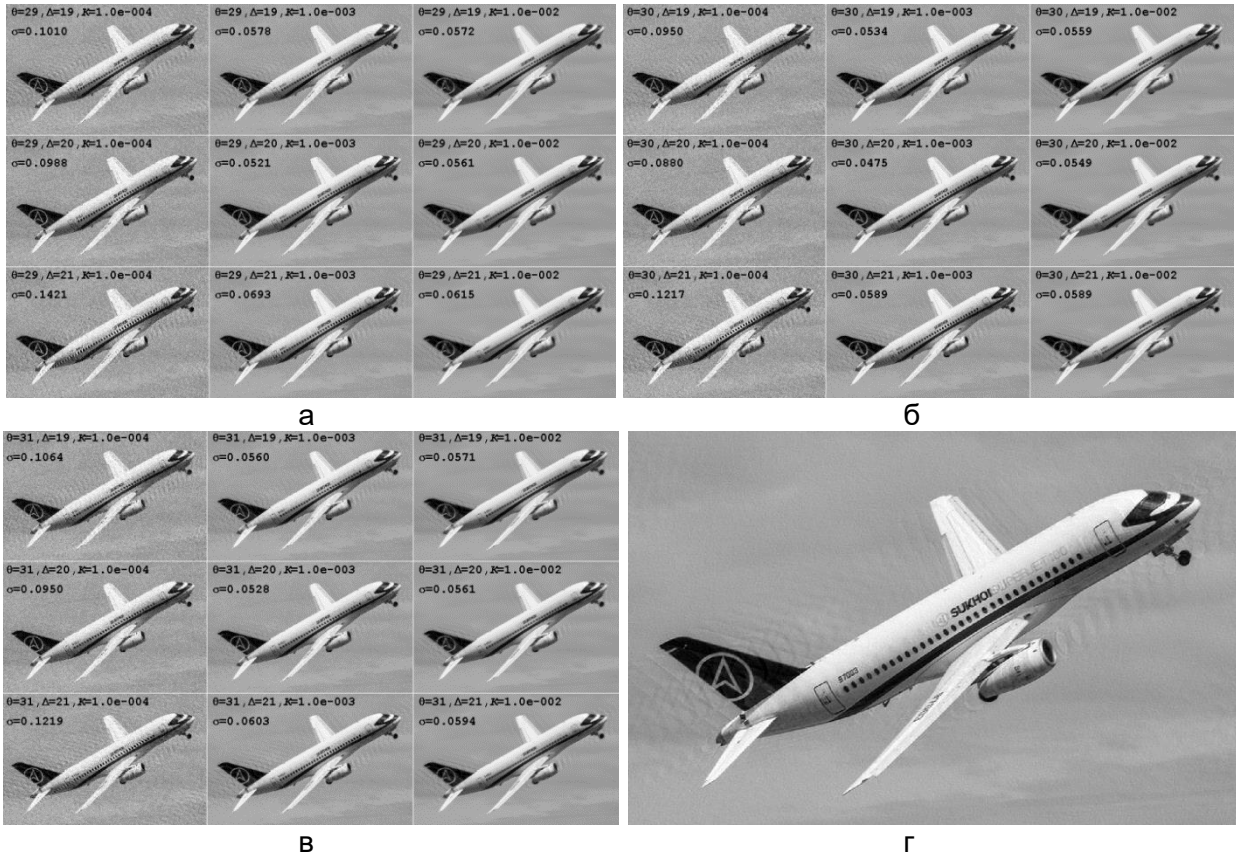


Рис. 2. Восстановление изображения при $\theta = 29°$ (а), $\theta = 30°$ (б), $\theta = 31°$ (в) и ряда значений Δ и K. Восстановленное изображение при $\theta = 30°$, $\Delta = 20$, $K = 10^{-3}$, σ=0,0475 (г)

По получении смазанного изображения $\tilde{g}(x, y)$ вычисляем $G(\omega_1, \omega_2)$ по m-функции fft2.m (один двумерный массив) и 27 спектров $W_K(\omega_1, \omega_2)$ решения согласно (4). В заключение вычисляем 27 возможных решений $w_K(x, y) = |F^{-1}(W_K(\omega_1, \omega_2))|$, используя m-функцию ifft2.m [7]. На рис. 2 даны результаты восстановления искаженного изображения.

Полагается, что восстановленные изображения выдаются на несколько экранов, и опытные диспетчеры визуально быстро выберут наилучшее изображение. Видно, что при $\theta = 30°$, $\Delta = 20$ пкс, $K = 10^{-3}$ получается наилучшее восстановление (рис. 2, г). На обработку изображений потребовалось 0,1 с. Заметим, что на кадрах приведены также значения относительной погрешности σ восстановления изображения [3, С. 139, 236].

На рис. 3 для сравнения с рис. 2, г, приведен результат восстановления изображения того же самолета, но методом РТ [6]. Видим, что метод ПФВ согласно (3) дает бóльшую погрешность σ восстановления изображения, чем метод РТ согласно (2).

Это можно объяснить большей упрощенностью метода ПФВ (см. константу K в спектре (3)) по сравнению с методом РТ (см. более сложное выражение $\alpha(\omega_1^2 + \omega_2^2)^p$ в (2)). Есть другие варианты метода ПФВ [3, С. 128], но они требуют большой дополнительной информации (спектров шума и решения).

Рис. 3. Изображение, восстановленное методом РТ при $\theta = 30°$, $\Delta = 20$, $\alpha = 10^{-3}$ (σ=0,0445)

Выводы. Предложен быстрый алгоритм восстановления смазанного изображения на основе решения двумерного интегрального уравнения (1) методом ПФВ. Особенностью алгоритма является то, что он использует заранее (до получения смазанного изображения) оценки значений угла смаза θ, величины смаза Δ и параметра метода *K*, а также предварительный расчет «заготовок» ряда спектральных функций для диапазонов возможных значений θ, Δ и *K*. Эти оценки могут быть получены путем предварительного машинного обучения. После получения смазанного изображения g будет получено несколько кадров восстановленного изображения, по которым диспетчеры сделают выбор решения, т.е. используется человеческий фактор. Сравнение ПФВ с методом РТ показывает, что ПФВ дает бóльшую погрешность (примерно на 3–10%), чем РТ, а также требует близкую компьютерную память. Оба метода могут быть использованы для восстановления в реальном времени изображений, смазанных, вообще говоря, под углом к горизонту, например, изображения самолета на взлете.

Литература

1. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах. – Новосибирск: Изд-во НГТУ, 2002. – 352 с.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. – 1072 с.
3. Сизиков В. С. Прямые и обратные задачи восстановления изображений, спектроскопии и томографии с MatLab. – СПб.: Лань, 2017. – 412 с.
4. Сизиков В.С., Кирьянов К.А., Экземпляров Р.А. Два быстрых алгоритма восстановления смазанных изображений // Изв. вузов. Приборостроение. – 2013. – Т. 56. – № 10. – С. 24–30.
5. Ильин А.К., Сизиков В.С. Восстановление смазанных изображений в реальном времени // Альманах научных работ молодых ученых ИТМО. – 2017. – Т. 2. – С. 100–104.
6. Сизиков В.С., Сергиенко А.А., Рущенко Н.Г. Новые быстрые алгоритмы восстановления смазанных под углом изображений объектов // Изв. вузов. Приборостроение. – 2019. – Т. 62. – № 1. – С. 69–77.
7. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. – М.: Техносфера, 2006. – 616 с.
8. Сизиков В.С. Спектральный способ оценки функции рассеяния точки в задаче устранения искажений изображений // Оптический журнал. – 2017. – Т. 84. – № 2. – С. 36–44.

Смогоржевский Глеб Александрович
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P41702
Направление подготовки: 09.04.02 – Мультимедиа в искусстве театра, кино и телевидения
e-mail: glebsmogor@gmail.com

Меженин Александр Владимирович
Год рождения: 1959
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: mejenin@mail.ru

УДК 004.921

МЕТОДЫ ГЕНЕРАЦИИ ГОРОДСКОЙ ИНФРАСТРУКТУРЫ В МЕДИЙНЫХ ПРИЛОЖЕНИЯХ

Смогоржевский Г.А., Меженин А.В.
Научный руководитель – к.т.н., доцент Меженин А.В.

В работе рассмотрены методы генерации, которые используются в процедурном моделировании для получения макетов виртуальных городов. Приведен обзор алгоритмов работы наиболее известных и популярных инструментов генерации виртуальных городов.
Ключевые слова: генерация, процедурно генерируемый контент, процедурное моделирование, генерация 3D-городов.

Методы и подходы к генерации. В работе рассмотрены некоторые методы и подходы процедурной генерации, которые используются или могут быть использованы при построении процедурной модели локации.

Диаграмма Вороного – это клеточные структуры, генерируемые точками, рассеянными на поверхности. Каждая точка находится в ячейке, где каждый край этой ячейки прижимается к краю ячейки ближайшей точки так, что каждое ребро равноудалено от двух ближайших точек. Метод может быть использован для создания клеточных геометрий. Например, данный метод используется для генерации карт-схем городов, где ячейки являются лотами районов города или ребра между ячейками используются в качестве дорожной сети [1].

Метод pointselection использует технику получения некоторой вводной информации от пользователя с последующей обработкой данной информации для генерации процедурного контента. Этот метод дает пользователю наиболее высокий уровень контроля, чем любой другой процедурный метод. По мнению автора, инструменты, использующие данный подход, являются наиболее актуальными для разработки процедурного контента на сегодняшний день [2].

Метод stackboxes используют для генерации болванок построек. Инициализируется первичный Box. Следующий Box создается в определенных пределах ребра или вершин предыдущей коробки. Цикл продолжается определенное количество итерацийй. Данный метод дает более жесткий контроль над правилами генерации построек и позволяет вносить изменения в получившуюся модель или саму систему генерации.

Макеты карт – это метод, использующий карты местности для генерации локаций. Карта может представлять собой фотографию, рисунок или 3D-объект, состоящий из районов и дорог между ними. Данный метод напрямую связан с pointselection [3].

Существующие решения. Далее рассмотрены некоторые существующие на данный момент решения, позволяющие получать процедурно сгенерированный город.

Citygen – система, которая использует процедурные методы для создания городских пейзажей. Система позволяет пользователю контролировать процесс генерации путем непосредственного управления параметрами алгоритма генерации через доступный и интуитивно понятный визуальный интерфейс, при этом результаты этих манипуляций вычисляются в реальном времени. Алгоритм генерации города делится на три этапа: построение графа дорог; генерация дорог; генерация зданий (рис. 1).

Сначала формируется первичная дорожная сеть. Дорожные сети представлены в виде неориентированных плоских графов. Затем при помощи алгоритма MinimumCycleBasis (MCB), из закрытых областей первичной дорожной сети формируются городские ячейки. После создания ячеек города начинается генерация второстепенных дорог с использованием алгоритма роста. Заключительным этапом является постройка сооружений вдоль дорог [4].

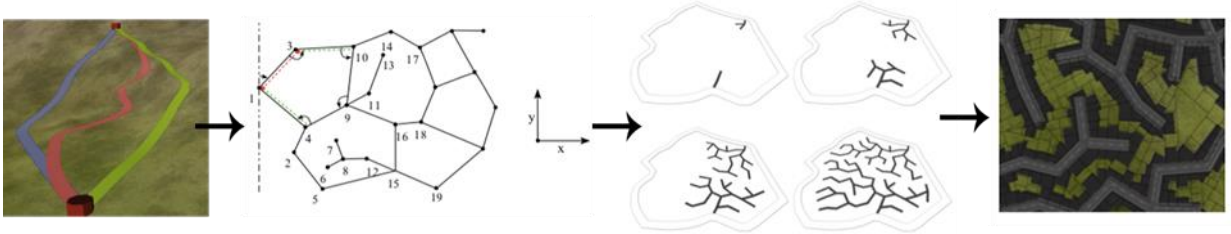


Рис. 1. Этапы работы Citygen

CityEngine – это программное обеспечение для 3D-моделирования, которое позволяет создавать огромные, интерактивные городские среды. Для создания города могут использоваться данных геоинформационной системы. Алгоритм работы состоит из пяти шагов (рис. 2). Сначала подготавливается базовая карта местности. Затем импортируются городской слой, добавляются улицы, городские кварталы и участки. После этого происходит конвертация 2D-геоданных в 3D, также добавляются крыши, машины, растения из 3D-ассетов. Далее запускается процесс текстурирования и добавления деталей. Заключительным этапом является экспорт получившегося макета [5].

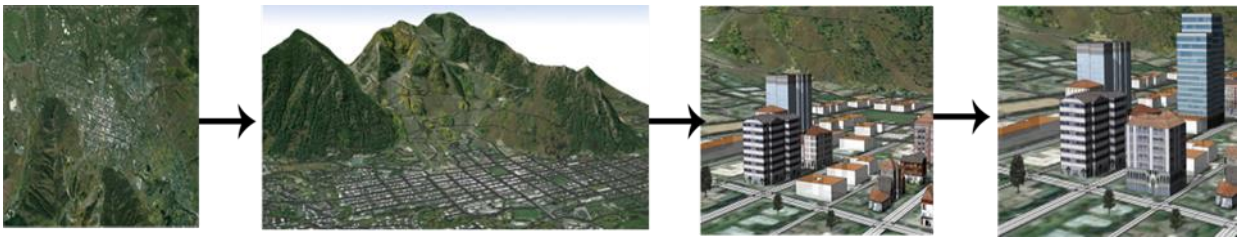


Рис. 2. Этапы работы CityEngine

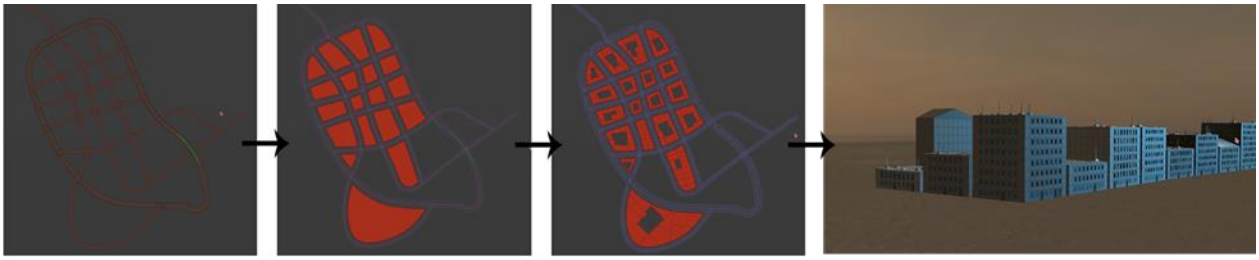


Рис. 3. Этапы работы MicroCity

MicroCity – это набор инструментов для SideFXHoudini, помогающий создавать процедурные здания и города. MicroCity представлен в виде набора ассетов, состоящих из нод, которые решают определенные задачи. Алгоритм генерации делится на четыре шага (рис. 3). Первый шаг заключается в генерации дорожной сети. После чего формируются основные площади, на которых будут сгенерированы здания. Затем эти площади разбиваются на лоты.

На полученных лотах происходит расстановка или генерация самих зданий. На взгляд автора это решение является наиболее приближенным и оптимальным для генерации контента [6].

Также необходимо сказать о Houdini. Houdini – это процедурная система, которая позволяет свободно создавать множество итераций и быстро обмениваться рабочими процессами с коллегами. В Houdini каждое действие хранится в ноде. Эти ноды объединяются в сети, которые можно настроить, чтобы уточнить результат. Способность узлов сохранять и передавать информацию в форме атрибутов по цепочке придает Houdini процедурный характер. Такая система позволяет экономить вычислительные ресурсы, и хорошо применима к методу PointSelection, так как на вход подаются только ссылки на префаб объектов, а не сами объекты. Это позволяет производить быстрые замены 3D-объектов в сложных и настроенных сценах. Благодаря этим особенностям на сегодняшний день Houdini является очень популярным инструментом для создания различного рода генераторов [7].

Заключение. Опираясь на полученные знания о применяемых при разработке методах генерации, можно прийти к выводу, что наиболее эффективными являются инструменты, комбинирующие несколько методов. Стоит отметить, что метод pointselection дает пользователю больше возможностей при манипуляции параметрами генерации. Еще одним плюсом данного метода является то, что его легко комбинировать с любым другим методом. В связи с этим планируется использовать данный метод при создании инструмента генерации и комбинировать его с методом макетов карт. Для генерации зданий планируется применить метод стековых ящиков, данный метод является довольно удобным для получения множества вариаций примитивов сооружений, с которыми можно работать. Средой разработки выбран Houdini. Данное программное обеспечение обладает большим рядом достоинств и идеально подходит для создания цифровых ассетов, реализующих множество процедурных задач.

Литература

1. Меженин А.В. Виртуальные 3D среды как средство верификации и тестирования при проектировании // Приоритетные научные направления: от теории к практике. – 2016. – № 21. – С. 105–110.
2. Безбах Ю.И., Меженин А.В. Оценка точности компьютерного моделирования освещения // Альманах научных работ молодых ученых Университета ИТМО. – 2016. – Т. 1. – С. 122–124.
3. SohamRamteke. Interactive Tool for Procedural City Generation in Houdini [Электронный ресурс]. – Режим доступа: https://nccastaff.bournemouth.ac.uk/jmacey/MastersProjects/MSc11/Soham/SohamRamteke_Thesis.pdf (дата обращения: 04.01.2019).
4. Kelly G. Hugh McCabe. Citygen: An Interactive System for Procedural City Generation [Электронный ресурс]. – Режим доступа: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.688.7603&rep=rep1&type=pdf (дата обращения: 04.01.2019).
5. CityEngine [Электронный ресурс]. – Режим доступа: https://www.esri.com/en-us/arcgis/products/esri-cityengine/overview (дата обращения: 04.01.2019).
6. MicroCity [Электронный ресурс]. – Режим доступа: https://sites.google.com/view/microcitygenerator/release-notes (дата обращения: 04.01.2019).
7. Baidachnyi D., Stafford L., Tokarev K. The Art of Technical Production in Games [Электронный ресурс]. – Режим доступа: https://80.lv/articles/the-art-of-technical-production-in-games/ (дата обращения: 04.01.2019).

Степанов Максим Юрьевич
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4220
Направление подготовки: 09.04.02 – Автоматизация и управление
в образовательных системах
e-mail: maximostepanov@gmail.com

Русак Алёна Викторовна
Год рождения: 1979
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: alena@cde.ifmo.ru

УДК 004.42

МОДЕЛЬ ПРОГНОЗИРОВАНИЯ НАЧАЛЬНОГО УРОВНЯ ИСПЫТУЕМОГО В ЭЛЕКТРОННЫХ СИСТЕМАХ ТЕСТИРОВАНИЯ

Степанов М.Ю.
Научный руководитель – к.т.н., доцент Русак А.В.

В работе рассмотрено построение и применение модели прогнозирования начального уровня испытуемого на основе его индивидуальных особенностей и результатов обучения для формирования адаптивной траектории тестирования.
Ключевые слова: электронное обучение, байесовская сеть доверия, прогнозирование, адаптивное тестирование, контроль знаний, теорема Байеса.

Электронные системы тестирования позволяют проводить контроль знаний учащихся в компьютеризированном виде, что упрощает процесс проверки и оценивания полученных результатов тестов. Они позволяют создавать как классические тесты с заранее определенным количеством вопросов и одинаковыми тестовыми вариантами для всех групп обучаемых, так и адаптивные, в которых вопросы разбиты по уровням сложности и предъявляются в зависимости от результата ответа на текущий уровень сложности тестовых заданий, следовательно, траектория и продолжительность процесса тестирования индивидуальны.

У классических и адаптивных тестов имеется общий недостаток – не учитывается начальный уровень знаний испытуемого. В первом случае это выражается в предъявлении одного и того же набора тестовых заданий для обучаемых разного уровня подготовки, во втором – точность результатов, как правило, зависит от количества предъявленных вопросов (больше – лучше). В классическом подходе можно составить несколько вариантов и присвоить их каждому испытуемому в соответствии с субъективной оценкой уровня его знаний, в адаптивном – добавить калибровочный блок для определения начального уровня сложности заданий, однако, это приводит к увеличению продолжительности тестирования. Прогнозирование начального уровня испытуемого позволит избежать субъективности выбора варианта при классическом подходе и дополнительных вопросов при адаптивном.

В данной работе рассмотрена модель прогнозирования начального уровня знаний испытуемого с применением байесовской сети доверия.

Определение исходного уровня испытуемого перед началом тестирования может проводиться по различным параметрам. Выявление таких факторов и степень их влияния

затронуты в большом количестве работ [1–3]. В результате для построения модели прогнозирования были выделены две группы факторов:

1. учебные:
– успеваемость по дисциплине;
– результаты тестирования по дисциплине;
– успеваемость по смежным дисциплинам;
– результаты тестирования по смежным дисциплинам;
2. внешние:
– комфортность условий проживания;
– способ подготовки к занятиям;
– наличие работы;
– участие в олимпиадах, грантах;
– студенческая активность;
– наличие достаточного количества времени на учебу.

На основании отобранных факторов было произведено построение байесовской сети доверия (БСД) [4]. БСД – направленный ациклический граф, состоящий из множества переменных и их вероятностных зависимостей по Байесу. Иногда ее называют причинно-следственной сетью, в которой случайные события соединены причинно-следственными связями. Устанавливая соответствующие значения параметров сети, можно произвести прогнозирование начального уровня знаний испытуемого перед началом процесса тестирования. На рисунке представлена разработанная БСД.

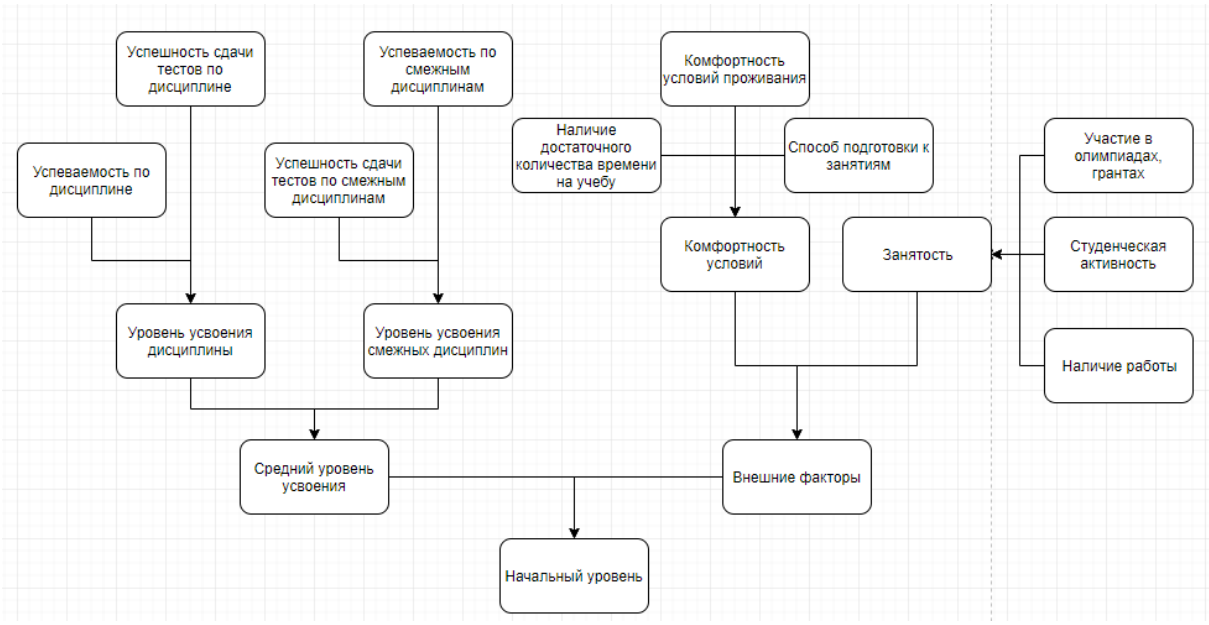


Рисунок. БСД для прогнозирования начального уровня испытуемого

Факторы, оказывающие влияние на испытуемого и его начальный уровень, являются корневыми (первопричинами) в построенной сети. Помимо этого, были определены промежуточные вершины для выявления общего влияния отдельных наборов причин в совокупности.

«Средний уровень усвоения» позволяет прогнозировать глубину и качество владения испытуемым учебным материалом. Она является дочерней по отношению к вершинам, позволяющим оценить в совокупности уровни усвоения по дисциплине тестирования и смежным ей: «уровень усвоения дисциплины», «уровень усвоения смежных дисциплин». Данная вершина позволяет сделать вывод об общей успеваемости испытуемого по дисциплине тестирования. В зависимости от значений родителей может принимать следующие состояния: «Высокий», «Средний», «Низкий».

Вершина «Уровень усвоения дисциплины» имеет две родительские вершины:

1. успеваемость по дисциплине;
2. успешность сдачи тестов по дисциплине.

«Успеваемость по дисциплине» вычисляется на основе данных об успеваемости студента по дисциплине. Данные берутся из электронной информационно-образовательной среды университета. Принимает состояния: «Высокая», «Средняя», «Низкая». Рассчитывается по формуле:

$$\frac{N}{N_{\max}} \times 100\%, \quad (1)$$

где N – текущее количество баллов; $N_{\max}$ – максимально возможное количество баллов.

Состоянию «Высокая» соответствует значение из интервала 80–100%, «Средняя» – 60–79%, «Низкая» – менее 60%.

«Успешность сдачи тестирования по данной дисциплине» вычисляется на основе количества баллов, полученных по результатам тестирования. Принимает состояния: «Высокая», «Средняя», «Низкая». Данные берутся из электронного журнала, применяемого в университете. Для определения состояния необходимо вычислить процентное соотношение количества набранных баллов за все тесты дисциплины к максимально возможному их количеству (используется формула (1)). В этом случае состоянию «Высокая» будет соответствовать от 80 до 100% набранных баллов от максимального, «Средняя» – 60–79%, «Низкая» – до 60%.

В результате можно сделать предположение об общей успеваемости испытуемого по дисциплине тестирования. В зависимости от значений родителей может принимать следующие состояния: «Высокая», «Средняя», «Низкая».

«Уровень усвоения смежных дисциплин» имеет две родительские вершины: «успеваемость по смежным дисциплинам», «успешность сдачи тестирования по смежным дисциплинам». Вершина и ее родители имеют состояния идентичные «уровню усвоения дисциплины» и «успеваемости по дисциплине» с «успешностью сдачи тестирования по дисциплине» соответственно.

«Занятость» может принимать следующие состояния: «Высокая», «Низкая». Позволяет определить, насколько сильно загружен испытуемый внеучебной деятельностью. Имеет следующие родительские вершины:

– «Участие в олимпиадах, грантах». Принимает состояния: «Участвует», «Не участвует». Данные берутся на основе анкетирования перед началом тестирования. По умолчанию выставляется значение «Не участвует»;
– «Студенческая активность». Характеризует занятость и активность участия в университетских и иных мероприятиях. Принимает состояния: «Высокая», «Низкая». Данные берутся на основе опроса перед началом тестирования;
– «Наличие работы». Принимает состояния «Полный рабочий день», «Частичная занятость», «Временные проекты», «Нет». Данные берутся на основе опроса перед началом тестирования. В случае отсутствия информации принимается значение «Нет».

Вершина «комфортность условий» является обобщающей для родительских факторов. Может принимать одно из состояний: «комфортные всегда», «комфортные периодически», «дискомфортные». Выбор состояния основывается на значениях родительских вершин: «наличие достаточного времени на учебу», «комфортность условий проживания», «способ подготовки к занятиям».

«Наличие достаточного времени на учебу» имеет три состояния: «Достаточно всегда», «Достаточно периодически», «Не хватает». Выбор состояния вершины основывается на данных, полученных при анкетировании перед началом тестирования. «Достаточно периодически» является стандартным состоянием при отсутствии данных.

«Способ подготовки к занятиям» имеет три состояния: «Активный», «Средней активности», «Пассивный». В первом случае испытуемый проходит обучающие тесты, повторяет пройденный материал и выполняет задания как выдаваемые, так и задания необязательные для закрепления материала. Во втором случае подразумевается периодическое выполнение заданий и повторение пройденного материала. В третьем – редкое повторение пройденного материала. Данные извлекаются из анкетирования, предъявляемого испытуемым перед началом процесса тестирования.

Вершины «средний уровень усвоения» и «внешние факторы» позволяют сделать промежуточные выводы об общей успеваемости и усвоении учебного материала испытуемым и степени влияния внешних факторов. В первом случае возможны следующие состояния: «Высокий», «Средний», «Низкий»; во втором случае: «Высокая загруженность», «Низкая загруженность».

Результирующая вершина «Уровень знаний» может принимать следующие состояния: «Низкий», «Средний», «Высокий». Возможно введение промежуточных значений: «Ниже среднего», «Выше среднего». Прогнозирование начального уровня испытуемого осуществляется путем выбора наиболее вероятного состояния данной вершины. Это возможно благодаря тому, что байесовские сети доверия позволяют произвести прогнозирование следствия по наблюдаемым причинам. Устанавливая соответствующие значения параметров байесовской сети, можно спрогнозировать предполагаемый начальный уровень знаний студента, что позволит сразу предъявлять вопросы, сложность которых будет соответствовать предполагаемому уровню знаний испытуемого.

Для установления вероятностных значений вершин по ветви «Внешние факторы» было произведено анкетирование 75 обучающихся первого курса Университета ИТМО. Для выставления значений по ветки «Средний уровень усвоения» был произведен анализ успеваемости соответствующих студентов. В качестве основного предмета была выбрана «Информатика», смежными для нее стали «Математика», «Дискретная математика», «Программирование». Дальнейшая пролангация свидетельств позволяет определить совместную вероятность любой комбинации событий в сети.

При определении начального уровня обучающихся они были поделены на две подгруппы. Для первой подгруппы (40 из 75 студентов) определение происходило как на основе анализа успеваемости, так и анкетирования. Прогнозирование для второй подгруппы (30 из 75 студентов) происходило на основе только данных об успеваемости. В результате было получено следующее распределение по предполагаемым начальным уровням (табл. 1).

Таблица 1. Распределение по начальным уровням

Предполагаемый начальный уровень	Количество человек
Низкий	15
Ниже среднего	0
Средний	49
Выше среднего	0
Высокий	11

На основании полученных результатов был сделан вывод, что достаточно выделить три уровня испытуемых. Наличие промежуточных уровней («ниже среднего» и «выше среднего») не оказывает влияние на итоговое прогнозирование. В табл. 2 представлено соответствие предполагаемых начальных уровней испытуемых с начальными уровнями сложности заданий в адаптивном тестировании и сложностью варианта при классическом подходе.

Таблица 2. Соответствие начальных уровней и уровней сложности

Предполагаемый начальный уровень	Уровень сложности вопроса/варианта теста
Низкий	Облегченный
Ниже среднего	Легкий
Средний	Средний
Выше среднего	Повышенной сложности
Высокий	Сложный

Для проверки модели прогнозирования были спроектированы адаптивные тесты по «Информатике», основывающиеся на тестах по соответствующей дисциплине в электронной информационно-образовательной системе «AcademicNT». За основу были выбраны обучающие тесты по темам «Информация и способы ее представления» и «Алгоритмы и структуры данных». В результате был разработан банк тестовых вопросов с ранжированием каждого задания по пяти уровням сложности (от легкого до сложного).

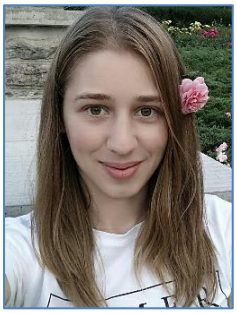
Далее было принято решение провести три тестирования со следующими условиями:

1. с применением модели прогнозирования на основе анализа успеваемости и внешних факторов (30 человек);
2. с применением модели прогнозирования только на основе анализа успеваемости (30 человек);
3. без применения модели прогнозирования с предъявлением среднего начального уровня сложности тестовых заданий (15 человек).

Первоначальные эксперименты показали работоспособность разработанной модели прогнозирования и возможность построения более гибких траекторий адаптивного тестирования с ее применением. В ходе дальнейших исследований планируются эксперименты по оптимизации и повышению точности работы представленной модели прогнозирования начального уровня испытуемого в электронных системах тестирования.

Литература

1. Harb N., El-Shaarawi A. Factors Affecting Students' Performance [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/24112529_Factors_Affecting_Students'_Performance (дата обращения: 04.01.2019).
2. Bagceci B., Kutlar E.L., Cinkara E. The Relationship between English and Math Success & some Variables at Freshman Level // Journal of Education and Practice. – 2014. – V. 5. – № 29. – P. 6–10.
3. Hijazi S.T., Raza Naqvi S.M.M. Factors affecting student’ performance // Bangladesh e-Journal of Sociology. – 2006. – V. 3. – № 1. – P. 1–10.
4. Сологуб Г.Б. Построение и использование байесовской сети для моделирования знаний студента в интеллектуальной системе тестирования // Компьютерные инструменты в образовании. – 2012. – № 2. – С. 40–48.



Ткешелашвили Нино Мерабиевна
Год рождения: 1992
Университет ИТМО, факультет программной инженерии и компьютерной техники, аспирант
Направление подготовки: 09.06.01 – Информатика и вычислительная техника
e-mail: ninomt@cs.ifmo.ru

Махновец Валерия Алексеевна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4212
Направление подготовки: 09.04.04 – Интеллектуальные информационные системы
e-mail: LerunchicM@gmail.com

Ткешелашвили Анна Мерабиевна
Год рождения: 1998
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р3410
Направление подготовки: 09.03.04 – Программная инженерия
e-mail: annamt@yandex.ru

Клименков Сергей Викторович
Год рождения: 1972
Университет ИТМО, факультет программной инженерии и компьютерной техники, ассистент
e-mail: serge@cs.ifmo.ru

УДК 004.822, 004.622

МОДУЛЬ КОНВЕРТАЦИИ ДОКУМЕНТОВ ДЛЯ СИСТЕМЫ АНАЛИЗА ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Ткешелашвили Н.М., Махновец В.А., Ткешелашвили А.М., Клименков С.В.
Научный руководитель – Клименков С.В.

Работа посвящена созданию модуля конвертации текстовых документов в унифицированный внутренний формат, предназначенный для обработки другими модулями системы семантического анализа. Особенностью формата является его ориентированность на структуру документа и семантические связи между его частями, а не на визуальное представление. Для чтения документов различных форматов модуль использует LibreOffice API.
Ключевые слова: конвертер документов, унифицированный формат, LibreOffice API, семантический анализ текстов, семантическая сеть.

Введение. Семантический анализ текстов на естественном языке является актуальным и развивающимся направлением исследований. Однако применение методов семантического

анализа требует унифицированных входных данных. Существует множество активно используемых форматов текстовых документов, таких как TXT, HTML, DOC, ODT, XLS, PDF и др. Большинство из них, помимо текста, содержит структуры данных и стили форматирования, которые также могут использоваться при семантическом анализе.

Целью представленной работы было создание модуля конвертации наиболее распространенных форматов текстовых документов в единый внутренний формат, предназначенный для обработки другими модулями системы семантического анализа текстов на естественном языке [1] с сохранением семантически значимой информации.

В основе системы лежит семантическая сеть, состоящая из понятий, связанных отношениями: синонимия, часть-целое, общее-частное. Помимо понятий и связей сеть содержит словоформы и экземпляры понятий, а также морфологические признаки словоформ. Ссылки на понятия используются во фреймах при поиске экземпляров понятий в тексте, за счет чего сеть автоматически пополняется из открытых источников, которые могут служить основой как новых узлов, так и экземпляров.

Модули системы используют семантическую сеть и механизм фреймов для решения практических задач, таких как: поиск и извлечение определенной информации из текста и электронных таблиц, автодополнение и др. Для того чтобы система могла работать с текстовыми файлами различных форматов, были разработаны внутренний формат документа и модуль конвертации из наиболее распространенных форматов документов в полученный формат.

Основной задачей конвертера являлось формирование унифицированного представления документа для различных форматов входных документов с сохранением информации о структурных связях частей документа и его форматировании, которая затем будет использоваться алгоритмами в процессе семантического анализа.

Архитектура конвертера. Входные документы представляют из себя набор взаимосвязанных компонентов. Например, простой текстовый документ может состоять из заголовка и нескольких параграфов, тогда как каждый параграф представлен предложениями, которые состоят из слов, цифр и знаков пунктуации. Табличные же документы состоят из ячеек, которые, в свою очередь, содержат текст или картинки. Таким образом, структура входного документа представлена деревом узлов, соединенных между собой связями разных типов. При этом каждый элемент связан как минимум со своим родителем и прямыми потомками, а также имеет связь «предыдущий» и «следующий», отражающую естественный порядок чтения.

При обработке конвертер разбивает исходный текст на токены. Разбиение осущетвляется отдельно разработанным токенизатором, работающим в режиме сервиса. Каждому токену ставится в соответствие отдельный узел в дереве документа. На данный момент используются несколько наиболее распространенных типов узлов, которые разделены на два класса: контейнеры (токен-группы), которые могут содержать в себе другие узлы, и токены. Для описания различных типов узлов документа реализован набор Java классов, базовым классом для которых является класс Element (рисунок).

Помимо узлов формат хранит связанные с ними стили, связи между узлами и допускает сохранение дополнительной метаинформации, такой как аннотация к узлам, позволяющей расширить систему.

Для преобразования исходных документов к одному из форматов разработанный конвертор использует API LibreOffice [2, 3]: ODT (для текстовых документов), ODS (для электронных таблиц) и ODG (для PDF). Затем каждый из трех форматов проходит индивидуальную обработку для извлечения текста и форматирования, после чего производится разбиение содержимого на токены и строится дерево документа.

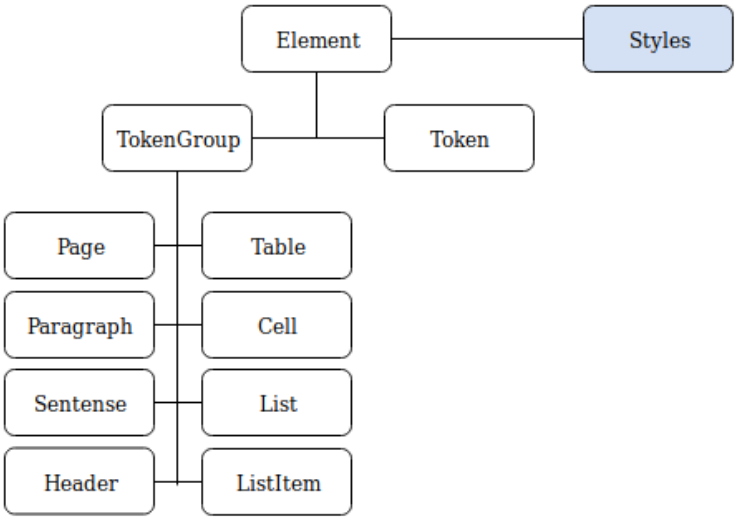


Рисунок. Диаграмма типов узлов итогового документа

Стили одного и того же фрагмента документа могут быть заданы различными способами. К примеру, ячейка таблицы ODS-документа имеет порядка 100 свойств, которые определяются выбранным стилем ячейки, свойствами самой ячейки и свойствами содержащегося в ней текста. В общей сложности документы рассматриваемых форматов используют 409 свойств, значения которых представлены объектами 42 различных классов. В свою очередь, свойства стилей могут быть составными, т.е. представлять собой объект с набором полей. Для таких свойств производится декомпозиция их на массив атомарных свойств, т.е. в набор пар «имя свойства» – «значение».

С целью сокращения объема информации из массива свойств удаляются некоторые свойства, не влияющие на визуальное отображение и смысловые отношения между элементами, а также все свойства, значение которых одинаково для всех элементов документа. Каждое свойство ассоциируется с конкретным элементом по уникальному идентификатору элемента. Полученный массив свойств хранится в корневом узле дерева элементов.

В ходе работы выполнен анализ представления семантической информации в различных типах документов, изучены современные программные средства для работы с офисными документами на языке Java, из которых выбран LibreOffice, и разработан модуль конвертера офисных документов в единый формат с сохранением семантических связей. Для хранения полученного документа и передачи его между модулями системы реализован экспорт в форматы XML и JSON.

Литература

1. Клименков С.В., Цопа Е.А., Жмылёв С.А., Покид А.В., Ткешелашвили Н.М. Метод быстрого поиска узлов семантической сети по точному совпадению словоформы // Изв. вузов. Приборостроение. – 2017. – Т. 60. – № 10. – С. 932–939.
2. LibreOffice 6.2 API Documentation [Электронный ресурс]. – Режим доступа: https://api.libreoffice.org/ (дата обращения: 04.01.2019).
3. The Apache OpenOffice API Project [Электронный ресурс]. – Режим доступа: http://www.openoffice.org/api/ (дата обращения: 04.01.2019).

Трушина Юлия Алексеевна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4270
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: yuliatr13@yandex.ru

Меженин Александр Владимирович
Год рождения: 1959
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: mejenin@mail.ru

УДК 004.92

МЕТОДЫ УМЕНЬШЕНИЯ ИСКАЖЕНИЙ ПРИ ЗАПЕКАНИИ КАРТ НОРМАЛЕЙ

Трушина Ю.А., Меженин А.В.
Научный руководитель – к.т.н., доцент Меженин А.В.

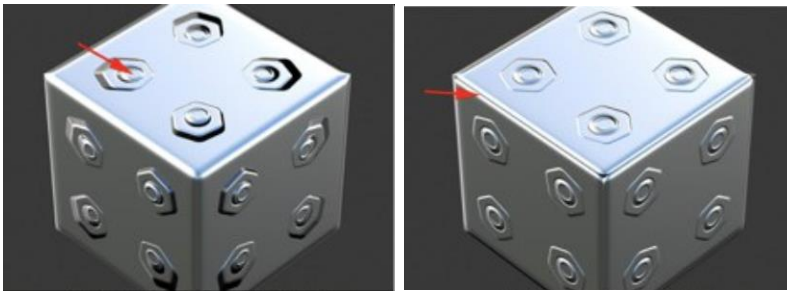
Этапы создания низкополигональной 3D-копии детализированной модели и последующего запекания всех деталей и неровностей поверхности с высокополигональной модели в карту нормалей являются одними из ключевых в процессе создания игровой модели. Для осуществления качественного запекания деталей на карте нормалей необходимо следовать ряду методов запекания карты нормалей для низкополигональной 3D-модели, которые описаны в работе.
Ключевые слова: нормаль, карта нормалей, запекание, Normal Map, Baking, детализация, LowPoly, HighPoly, 3D-модель.

Технология Normal Mapping, применяемая для симуляции детализации поверхности на низкополигональных (LP) 3D-моделях, является активно применяемой технологией, способствующей экономии ресурсов с точки зрения полигональных характеристик 3D-моделей и сохранению их детализированности и визуальной зрелищности [1].

При запекании поверхности HP-модели на ее LP-копию зачастую возникает проблема с имеющимися выпуклыми деталями: они выглядят искаженными и скошенными (деформированными) на запеченной NM. Причина возникновения таких ошибок лежит в особенности проецирования детализированной геометрии на низкодетализированную. При фиксации деталей высокополигональной (HP) модели в карте нормалей применяется проецирующая сетка (cage). Она строится вокруг исходной HP-модели и проецирует лучи, чтобы в результате перенести в Normal Map информацию о поверхности модели для освещенности LP-модели.

Для запекания детализации есть два типа проецирующих сеток: Average Normals и Explicit Normals.

При использовании Average Normals cage проецируется с усредненными нормалями, в то время как нормали LP-модели не учитываются. Положительным результатом сглаживания нормалей является отсутствие швов на местах фасок, даже когда применяются жесткие грани или группы сглаживания [2]. Однако в таком случае детали на карте нормалей деформируются (рис. 1, а).



а б

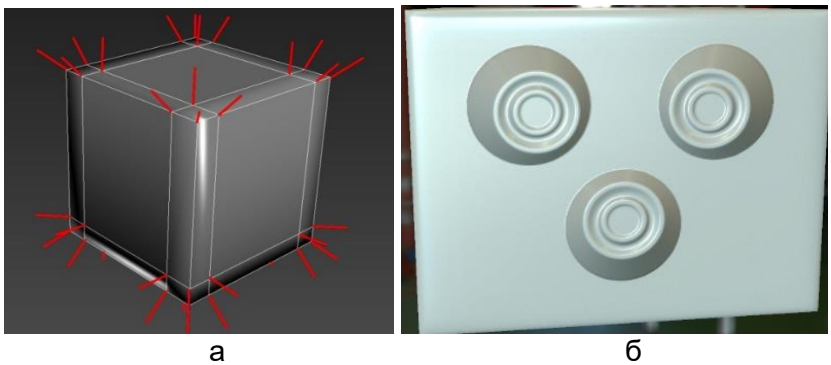
Рис. 1. Демонстрация искажения запеченных на LP-модели деталей (а) и запекания швов на LP-модели (б)

Второй вариант проецирования cage не игнорирует нормали LP-модели, а использует именно их для построения cage. Поэтому при использовании на ребрах жестких граней или на границах групп сглаживания на запеченной карте нормалей в местах фасок остаются швы, в то время как все детали, которые искажаются в первом способе, запекаются качественно (рис. 1, б).

Для решения описанной проблемы запекания деталей с искажениями можно выбрать один из нескольких алгоритмов работы:

1. добавление дополнительных поддерживающих ребер;
2. Mesh skewing;
3. применение инструмента Paint Skew в Marmoset Toolbag;
4. совмещение нескольких NM в графическом редакторе.

Для борьбы с некорректно запеченными деталями на плоскостях можно использовать первый метод добавления временной геометрии, т.е. слегка усложнить геометрию LP-модели, добавив ей поддерживающие грани. Дополнительные ребра на LP-модели помогают усреднить нормали вертексов – вершин полигональной сетки 3D-модели (рис. 2, а). Таким образом, на выровненной с их помощью поверхности в карте нормалей детали запекутся в необходимом направлении, не исказившись из-за усреднения нормалей LP-модели.



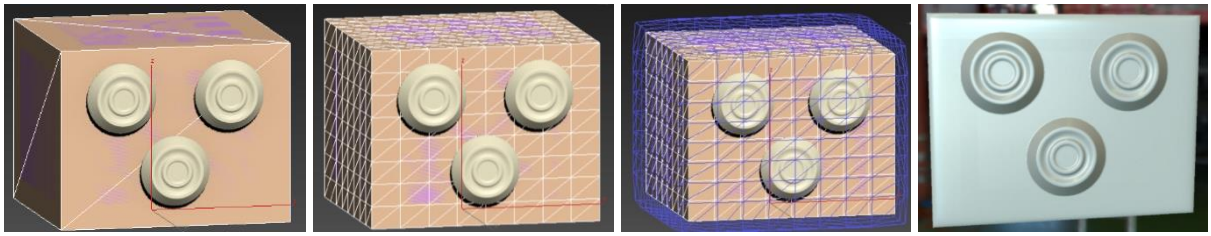
а б

Рис. 2. Добавление дополнительной геометрии, усредняющей вертексные нормали (а); искаженные детали на поверхности LP-модели (б)

После запекания NM на LP-модели можно удалить временно добавленную геометрию и применить к ней запеченную на усложненной модели карту.

Если первый метод оказался трудно применимым, можно использовать другой – Mesh skewing. На рис. 2, б, продемонстрирована проблема запекания деталей.

Чтобы исправить перенесенную на карте нормалей деформацию детализации, нужно триангулировать (разбить полигоны LP-модели на треугольники), после чего тесселировать модель, например, в 3ds Max применить модификатор Tesselate (рис. 3, а–в).



а б в г

Рис. 3. Триангуляция (а); тесселяция LP-модели (б); настройка ее проецирующей cage (в); отсутствие искажений на запеченных на LP-модели деталях (г)

После экспортирования модели в среду для запекания текстурных карт (например, Marmoset Toolbag) необходимо отрегулировать cage – расположить проецирующие лучи так, чтобы они фиксировали изгибы высокодетализированной геометрии – и запечь карту нормалей типа Normals Object. После чего преобразовать ее в карту типа Tangent. Для этого можно использовать конвертирование карт нормалей в программе xNormal [3]. После применения этого метода результат запекания деталей окажется корректным (рис. 3, г).

Третий способ избавления от деформаций на запеченных деталях в программе Marmoset Toolbag заключается в использовании функции рисования инструментом Paint Skew для исправления некорректного запекания деталей на карте нормалей. С его помощью с регулируемой степенью интенсивности можно смешивать нормали кейджа и поверхности объекта, рисуя черно-белой кистью на самой карте нормалей, исправляя, таким образом, нормали запекаемых деталей (рис. 4).

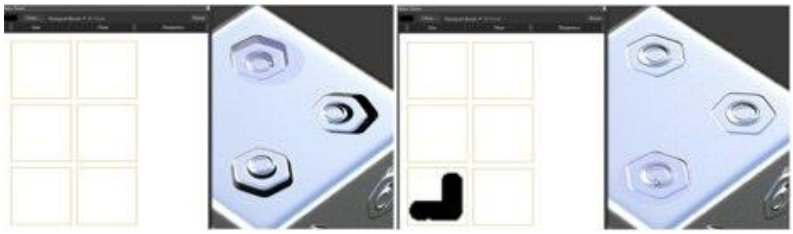


Рис. 4. Демонстрация ручного исправления нормалей на NM с помощью инструмента Paint Skew

Еще одним способом получения желаемого результата запекания деталей является использование запекания обоих типов проецирования cage (Average normals и Explicit normals) и последующее ручное редактирование по совмещению полученных результатов в графическом редакторе с помощью масок так, чтобы на финальной карте нормалей были и ровные детали, и гладкие перегибы граней без швов [4].

Описанные методы позволяют решить сложности, связанные с запеканием поверхности с HP-моделей на LP без искажений и деформации перенесенных деталей. В дальнейшем планируется продолжение рассмотрения методов борьбы с искажениями перенесенной детализации на низкополигональных 3D-моделях [5–7].

Литература

1. Меженин А.В. Методы и средства распознавания образов и визуализации: технология Normal Mapping. – СПб.: НИУ ИТМО, 2012. – С. 75–76.
2. Меженин А.В. Технологии разработки 3D-моделей. – СПб.: Университет ИТМО, 2018. – 100 с.
3. Смолин А.А., Жданов Д.Д., Потемин И.С., Меженин А.В., Богатырев В.А. Системы виртуальной, дополненной и смешанной реальности. – СПб.: Университет ИТМО, 2018. – 59 с.

4. Тозик В.Т., Меженин А.В. Метод геометрического упрощения 3D полигональных объектов // Научно-технический вестник информационных технологий, механики и оптики. – 2010. – № 3(67). – С. 81–85.
5. Tips for Creating Perfect Normal Maps Every Time [Электронный ресурс]. – Режим доступа: https://www.pluralsight.com/blog/tutorials/tips-creating-perfect-normal-maps-every-time/ (дата обращения: 04.01.2019).
6. O'Hare J. Unity3D Tangent Basis Calculator Plugin for xNormal [Электронный ресурс]. – Режим доступа: http://www.farfarer.com/blog/2012/06/12/unity3d-tangent-basis-plugin-xnormal/ (дата обращения: 04.01.2019).
7. Абрамова О.Ф., Книжко А.В. Исследование методов текстурирования ландшафта со сложным рельефом [Электронный ресурс]. – Режим доступа: https://novainfo.ru/article/pdf/?nid=8950 (дата обращения: 04.01.2019).

Урываев Кирилл Антонович
Год рождения: 1996
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P41212
Направление подготовки: 09.04.04 – Нейротехнологии и программная инженерия
e-mail: spore9@yandex.ru

Русак Алёна Викторовна
Год рождения: 1979
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: alena@cde.ifmo.ru

УДК 004.8

ОБЗОР МЕТОДОВ ПОВЫШЕНИЯ КАЧЕСТВА ПРОГНОЗИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Урываев К.А.
Научный руководитель – к.т.н., доцент Русак А.В.

В работе рассмотрены основные методы повышения качества, т.е. скорости и точности прогнозирования с помощью искусственных нейронных сетей. Данные параметры имеют критическую важность, так как во многих реальных практических задачах цена ошибки прогнозирования крайне велика. Искусственные нейронные сети хорошо справляются с задачей прогнозирования, но в некоторых случаях они могут давать неудовлетворительный результат. Однако существует ряд методов, направленных на повышение качества обучения искусственных нейронных сетей.
Ключевые слова: искусственные нейронные сети, машинное обучение, прогнозирование, нелинейная регрессия, инициализация Ксавьера.

Прогнозирование – нелинейная и часто достаточно труднореализуемая задача, очевидно, являющаяся актуальной на данный момент, находит свое применение как в чисто научных областях (физика, химия, биология и т.д.), так и в непосредственно прикладных (маркетинг, статистика и т.д.). Суть задачи прогнозирования состоит в предсказании будущей реакции системы по ее предшествующему поведению. Данная задача хорошо решается с помощью искусственных нейронных сетей (ИНС), в частности рекуррентной ИНС. Для ИНС прогнозирование – задача нелинейной регрессии. Обладая информацией о значениях переменной x в моменты, предшествующие прогнозированию $x(k–1)$, $x(k–2)$, …, $x(k–N)$, сеть вырабатывает решение, каким будет наиболее вероятное значение последовательности $\bar{x}(k)$ в текущий момент k. Для адаптации весовых коэффициентов сети используются фактическая погрешность прогнозирования $\varepsilon = x(k) - \bar{x}(k)$ и значения этой погрешности в предшествующие моменты времени [1].

В задачах с небольшим набором исходных данных или большой алгоритмической сложностью решение задачи прогнозирования может давать неудовлетворительные результаты или приводить к большим объемам вычислений. При этом с учетом продолжающейся автоматизации, в системах, чья функциональность охватывает важные и опасные сферы, цена ошибки прогнозирования имеет критическую важность. Для решения

или смягчения данной проблемы существуют различные методы повышения качества прогнозирования, которые и рассмотрены в данной работе.

Инициализация Ксавьера заключается в генерировании случайных начальных весов связей, основанных на числе входных и выходных связей данного нейрона. Данный метод позволяет ускорять обучение ИНС [2] и повышает качество прогнозирования, так как при нескольких обучающих итерациях, ИНС не попадает в один и тот же локальный минимум. Начальный вес связи рассчитывается по формуле:

$$W_i = U\left(-\frac{\sqrt{6}}{\sqrt{Count_{in}+Count_{out}}}, \frac{\sqrt{6}}{\sqrt{Count_{in}+Count_{out}}}\right), \quad (1)$$

где Wi – вес связи нейрона; U – равномерное распределение; $Count_{in}$ – число входов нейрона; $Count_{out}$ – число выходов нейрона.

Перемешивание данных – это расположение наборов данных в обучающей и тестовой выборке случайным образом. Данный метод позволяет избавиться от зависимости результата от последовательности данных. Однако данный прием не всегда подходит для задач прогнозирования [3], так данные могут иметь зависимость от последовательности.

Использование динамического коэффициента обучения позволяет варьировать шаг обучения в зависимости от ситуации. Например, если был достигнут локальный минимум при шаге 1, то при шаге 0,1 локальный минимум будет установлен более точно [4].

Для изменения коэффициента необходимо иметь критерий, самым простым и очевидным является использование функции стоимости. Это подразумевает, что, когда сеть начнет расходиться, можно попробовать уменьшить шаг, чтобы достигнуть более точного значения минимума. Либо же наоборот, можно увеличить шаг, если, например, мы уверены, что сеть попала в локальный минимум, и ей следует искать решение дальше.

Пример использования динамического коэффициента обучения представлен на рис. 1, где синие стрелки – это пример слишком мелкого коэффициента обучения, в то время как красные стрелки – пример слишком большого значения. Зеленые стрелки показывают преимущество динамического коэффициента обучения.

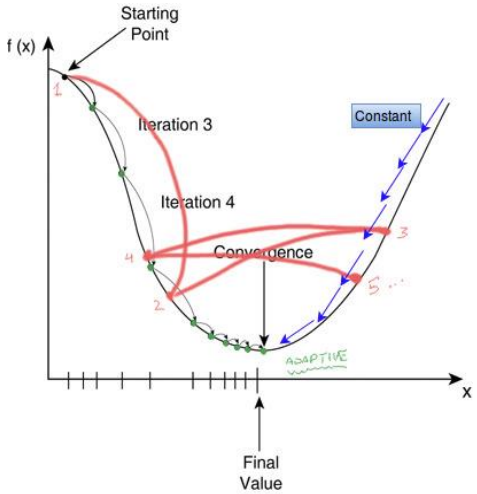


Рис. 1. Использование динамического коэффициента обучения

Суть регуляризации заключается во введении дополнительного сглаживающего коэффициента, который не позволяет весам приобретать очень большие значения, т.е. решает задачу переобучения нейронной сети.

Значение регуляризации можно найти по формуле:

$$regularization = \frac{\lambda}{2m}\sum_{j=1}^{n} W_j^2, \quad (2)$$

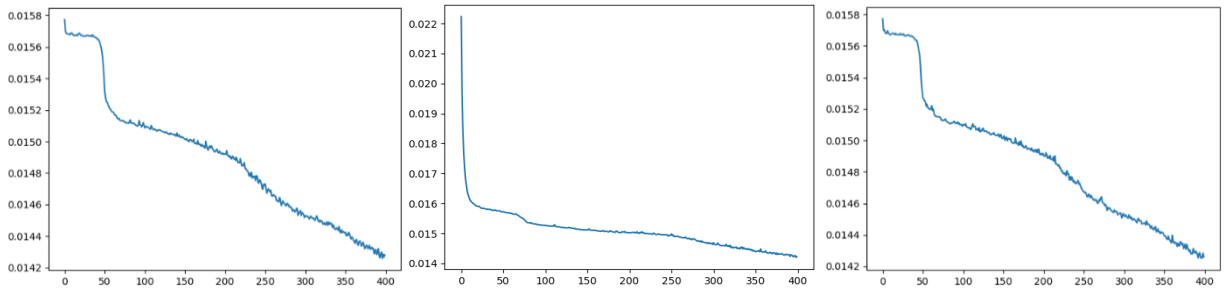
где m – размер выборки; λ – коэффициент регуляризации; n – число связей нейрона; W – вес связи.

Метод прунинг заключается в приравнивании к 0 самых малых относительно остальных связей. Данный метод является более спорным, так как часто ухудшает результат, так как может обнулять важные связи, но при этом в больших сетях позволяет существенно облегчить вычисления [5]. Также для реализации прунинга необходимо использовать разреженные матрицы, т.е. матрицы с большим количеством нулей.

Для проверки описанных методов были проведены тесты на задаче Flare из набора Proben1. В данной задаче имеется 24 входа и три выхода, строк данных – 1066. Для реализации проверки был выбран фреймворк Keras на TensorFlow.

Для сравнения была решения задача прогнозирования с помощью рекуррентной ИНС без использования методов повышения качества прогнозирования. На рис. 2, а, показан график скорости обучения такой сети, где за 400 итераций сеть так и не обучилась до конца.

На рис. 2, б, в, приведены графики скорости обучения рекуррентной сети при применении инициализации Ксавьера (рис. 2, б) и динамического коэффициента обучения (рис. 2, в). Как видно из рисунков, динамический коэффициент обучения не оказал влияния на скорость обучения, в то время как инициализация Ксавьера изменила его характер.



а б в

Рис. 2. Скорость обучения: изначальной ИНС (а); с инициализацией Ксавьера (б); с динамическим коэффициентом обучения (в)

Наилучший результат дает композиция рассмотренных методов. При совместном применении нейронная сеть обучается уже к 20 итерации, что отображено на рис. 3.

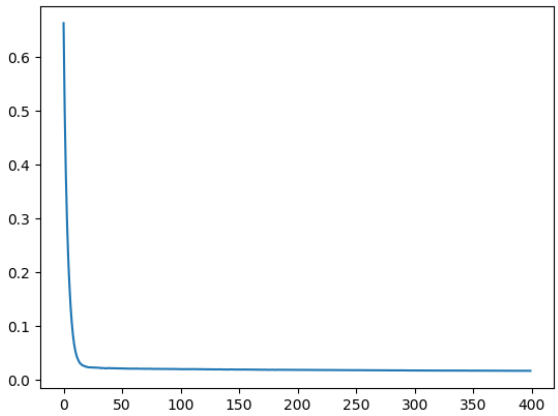


Рис. 3. Скорость обучения ИНС с использованием всех методов

В таблице представлен результат сравнения точности рассмотренных методов. За функцию потерь в данном случае была принята абсолютная квадратичная ошибка, так как это задача регрессии.

Таблица. Точность ИНС

Вид РНН	Функция потерь	Точность
Изначальная РНН	0,01626	87,5%
Инициализация Ксавьера	0,01617	91,25%
Динамический коэффициент обучения	0,01607	78,75%
Регуляризация	0,01556	83,75%

Вид РНН	Функция потерь	Точность
Перемешивание данных	0,01552	88,75%
Композиция методов	0,01484	97,8125%

Как видно из таблицы, некоторые методы улучшали результат, некоторые даже ухудшали, но наилучший результат был получен при одновременном использовании всех методов.

Литература

1. Осовский С. Нейронные сети для обработки информации / Пер. с польского И.Д. Рудинского. – 2-е изд. – М.: Финансы и статистика, 2017. – 448 с.
2. LeCun Y., Bottou L., Orr G., Muller K. Efficient BackProp [Электронный ресурс]. – Режим доступа: http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf (дата обращения: 04.01.2019).
3. Brownlee J. How to Backtest Machine Learning Models for Time Series Forecasting [Электронный ресурс]. – Режим доступа: https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/ (дата обращения: 04.01.2019).
4. Moreira M., Fiesler E. Neural Networks with Adaptive Learning Rate and Momentum Terms [Электронный ресурс]. – Режим доступа: https://infoscience.epfl.ch/record/82307/files/95-04.pdf (дата обращения: 04.01.2019).
5. Gildenblat J. Pruning deep neural networks to make them fast and small [Электронный ресурс]. – Режим доступа: https://jacobgil.github.io/deeplearning/pruning-deep-learning (дата обращения: 04.01.2019).

Фишман Владислав Олегович
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4272
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: fanmeeee@gmail.com

Меженин Александр Владимирович
Год рождения: 1959
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: mejenin@mail.ru

УДК 004.5

ДИЗАЙН ИНТЕРФЕЙСОВ АВТОМОБИЛЬНЫХ МУЛЬТИМЕДИА СИСТЕМ

Фишман В.О., Меженин А.В.
Научный руководитель – к.т.н., доцент Меженин А.В.

В работе рассмотрены существующие на данный момент интерфейсы систем мультимедиа автомобилей. Проведен анализ аналогов. Разработан предлагаемый интерфейс системы.
Ключевые слова: интерфейс системы мультимедиа, автомобильная мультимедиа система, анализ аналогов мультимедиа систем.

Развитие бортового компьютера автомобиля началось, в том числе с развитием в индустрии авиации. Дизайнеры автомобилей копировали бортовые элементы, которых в транспортном средстве было огромное количество.

В современных транспортных средствах функции бортового компьютера распределены на комбинации приборов (щиток приборов) и на системе мультимедиа (головное устройство мультимедиа) [1–3]. Взаимодействие водителя с системой мультимедиа должно происходить эффективно, водитель должен как можно меньше времени отвлекаться от дороги, потому что это вопрос безопасности [4, 5]. В связи с этим целью работы являлся принцип проектирования пользовательского интерфейса системы мультимедиа автомобиля, для повышения эффективности взаимодействия водителя с ней.

Были рассмотрены различные системы мультимедиа автомобильных марок (KIA, Suzuki, Infiniti, Porshe, Mersedes, Yaguar, Volvo, Volksvagen, Mazda, Toyota и др.). Все модели автомобилей 2018/2019 модельного года. Рассмотрим некоторые из них более подробно.

На рис. 1, а, представлен интерфейс системы мультимедиа автомобиля Suzuki SX4.

Достоинства интерфейса – достаточно высокая контрастность элементов. К недостаткам можно отнести: много пустого места на экране, нечитабельный мелкий шрифт, мелкие иконки.

На рис. 1, б, представлен интерфейс системы мультимедиа автомобиля Mazda CX5.

Достоинствами интерфейса является: минимализм, крупные иконки, крупный текст. К недостаткам интерфейса можно отнести: работа с телефонной книгой через Bluetooth.

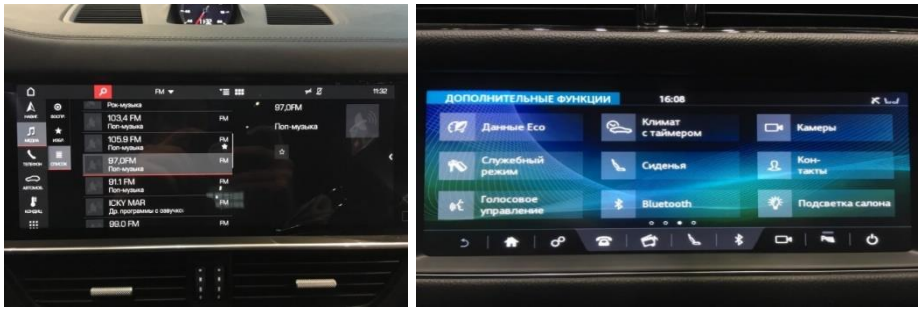
а б

Рис. 1. Интерфейс системы мультимедиа: Suzuki SX4 (а); Mazda CX5 (б)

На рис. 2, а, представлен интерфейс системы мультимедиа автомобиля Porsche Cayene, достоинствами которого является: приятная анимация, светлые элементы на темном фоне, понятные иконки, голосовое управление. К недостаткам интерфейса можно отнести: много элементов на экране, сложность в нахождении нужной функции, сложно найти нужную, присутствие элементов без которых не потеряется смысл.

На рис. 2, б, представлен интерфейс системы мультимедиа автомобиля Yaguar XF.

Достоинствами интерфейса является: читабельный текст, понятные иконки, голосовое управление. К недостаткам интерфейса можно отнести: много элементов на экране, много пустого места на экране, сложность в нахождении нужной функции, слабая приоритезация функций.



а б

Рис. 2. Интерфейс системы мультимедиа: Porsche Cayene (а); Yaguar XF (б)

Для того чтобы устранить выявленные недостатки существующих систем мультимедиа автомобилей, было предложено: использовать пространство экрана по максимуму, убрать или скрыть ненужные функции, выделить приоритеты функций и сделать кнопки разных размеров, не отображать все элементы в одной цветовой гамме, сделать шрифт крупнее, сделать иконки крупнее, увеличить контрастность.

На рис. 3 представлен макет предлагаемого интерфейса системы мультимедиа автомобиля. Макет интерфейса был создан в программах Adobe Illustrator, Adobe XD.

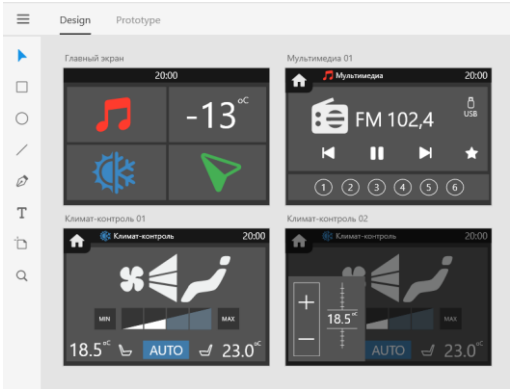


Рис. 3. Макет предлагаемого интерфейса

В дальнейшем планируется разработать альтернативный предлагаемый интерфейс системы мультимедиа и провести эксперимент, в ходе которого будет производиться оценка времени выполнения различных сценариев при использовании интерфейсов.

Литература

1. Сверчков Д.С. Разработка человеко-машинного интерфейса и его применение в системах управления // Труды Крыловского государственного научного центра. – 2018. – Вып. 1. – С. 184–190.
2. ГОСТ Р 51341-99. Безопасность машин. Эргономические требования по конструированию средств отображения информации и органов управления. Часть 2. Средства отображения информации. – Введен 22.11.1999. – М.: Госстандарт России, 1999. – 2 с.
3. Strayer D.L. Prof david strayer says technology that doesn't mix with driving continues to increase in many new vehicles [Электронный ресурс]. – Режим доступа: https://unews.utah.edu/caution-ahead-the-growing-challenge-for-drivers-attention/ (дата обращения: 03.12.2018).
4. Смолин А.А., Жданов Д.Д., Потемин И.С., Меженин А.В., Богатырев В.А. Системы виртуальной, дополненной и смешанной реальности. – СПб.: Университет ИТМО, 2018. – 59 с.
5. Меженин А.В. Исследование систем стереозрения средствами виртуального моделирования // Вопросы образования и науки: теоретический и методический аспекты. Сб. науч. тр. по мат-лам Междунар. заоч. науч.-практ. конф. – 2012. – Ч. 4. – С. 96–98.

Фролов Артём Игоревич
Год рождения: 1993
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4217
Направление подготовки: 09.04.04 – Программная инженерия
e-mail: frolov_a@auca.kg

Симоненко Зинаида Григорьевна
Год рождения: 1950
Университет ИТМО, факультет программной инженерии
и компьютерной техники, ст. преподаватель
e-mail: zgsim@yandex.ru

УДК 004.65

АНАЛИЗ АЛГОРИТМА ГРАДИЕНТНОГО БУСТИНГА ДЛЯ ЦЕЛЕЙ ПРОГНОЗИРОВАНИЯ

Фролов А.И.
Научный руководитель – Симоненко З.Г.

В работе рассмотрен метод градиентного бустинга и возможности его применения для целей прогнозирования. Наиболее точные и приближенные значения прогнозов на конкретные услуги значительно снижают риски с убытками и излишними ресурсами. Сложность прогнозирования заключается, во-первых, в подборе и правильном использовании существующей модели, описываемой алгоритмами машинного обучения. Во-вторых, модель должна соответствовать степени точности, которая определяет практическую применимость модели. Необходимым условием также является соответствие или несоответствие полученных данных машинного обучения с реальными данными предприятий. В настоящее время очень популярна технология градиентного бустинга. Этот метод незаменим там, где необходимо улучшить существующую модель путем наложения бустинга.
Ключевые слова: градиентный бустинг, машинное обучение, прогнозирование, математическая модель, деревья решений.

Введение. Машины градиентного бустинга являются чрезвычайно популярным алгоритмом машинного обучения, который доказал свою эффективность во многих областях. В то время как случайные леса строят ансамбль глубоких независимых деревьев, бустинговые машины строят ансамбль мелких и слабых последовательных деревьев, каждое дерево изучает и совершенствует предыдущее. В сочетании эти многочисленные слабые последовательные деревья создают мощный «ансамбль», который зачастую трудно превзойти с помощью других алгоритмов [1].

Целью исследования являлась разработка прогностической модели на основе алгоритма градиентного бустинга для анализа данных и прогнозирования. В ходе исследования алгоритмов для целей прогнозирования были выполнены следующие задачи:

- исследование современных высокоэффективных методов прогнозирования данных и обоснование выбора технического решения;
- разработка модели, удобной для построения прогнозов на разнотипных данных;
- выбор оптимальной модели, обеспечивающей точную прогностическую вероятность.

Исследование градиентного бустинга. Многие контролируемые модели машинного обучения основаны на одной прогностической модели (т.е. линейной регрессии, штрафных моделях, наивных байесовских алгоритмах, машинах опорных векторов). Альтернативно, другие подходы, такие как пакетирование и случайные леса, основаны на идее построения ансамбля моделей, где каждая отдельная модель предсказывает результат, а затем ансамбль просто усредняет предсказанные значения. Семейство методов бустинга основано на другой, конструктивной стратегии формирования ансамбля.

Основная идея бустинга заключается в последовательном добавлении новых моделей в ансамбль. На каждой конкретной итерации новая слабая модель базового ученика обучается в отношении ошибки всего изученного ансамбля.

Последовательное обучение по отношению к ошибкам: бустинговые деревья растут последовательно; каждое дерево выращивается с использованием информации из ранее выращенных деревьев. Данная модель представлена на рис. 1.

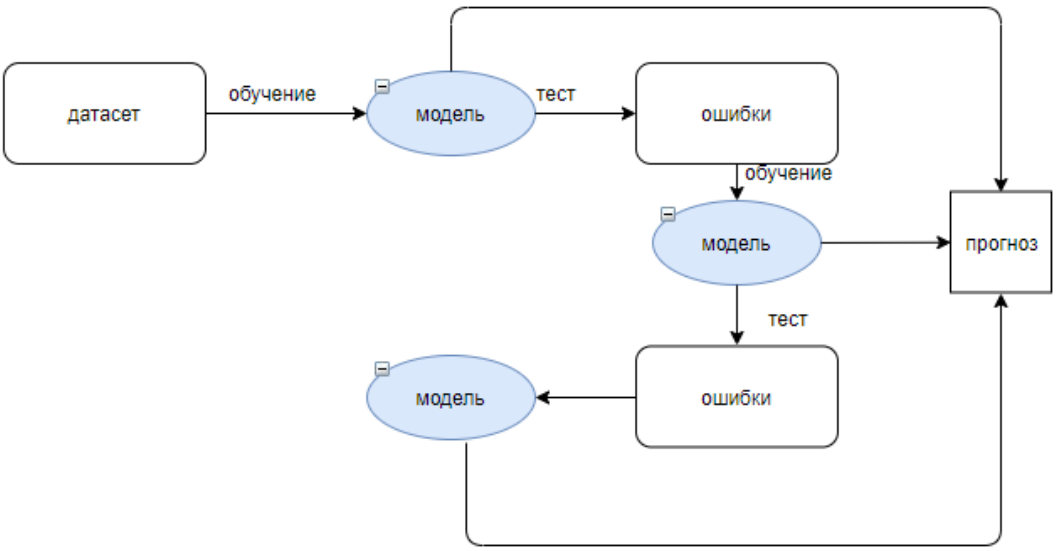


Рис. 1. Метод последовательного ансамбля

Обобщая все вышесказанное, бустинг – это фреймворк, который итеративно улучшает любую слабую модель обучения. Многие приложения повышения градиента позволяют подключать различные классы слабых учеников, однако на практике ускоренные алгоритмы почти всегда используют деревья решений в качестве базового обучающегося. Следовательно, в большинстве случаев бустинг рассматривается в контексте деревьев регрессии [2].

Слабая модель – это модель, частота ошибок которой лишь немного лучше, чем случайное угадывание. Идея бустинга заключается в том, что каждая последовательная модель строит простую слабую модель, чтобы немного улучшить оставшиеся ошибки [3]. Что касается деревьев решений, мелкие деревья представляют слабого ученика. Обычно используются деревья только с 1–6 разбиениями. Объединение многих слабых моделей (по сравнению с сильными) имеет несколько преимуществ:

– построение слабых моделей вычислительно дешево;
– повышение точности: слабые модели позволяют алгоритму учиться медленно;
– вносить незначительные коррективы в новые области, где он неэффективен;
– в целом статистические подходы, которые учатся медленно, имеют тенденцию работать хорошо;
– предотвращает переобучение: благодаря внесению только небольших постепенных улучшений с каждой моделью в ансамбле, это позволяет останавливать процесс обучения, как только переобучение было обнаружено (обычно с помощью перекрестной проверки).

Формализация модели градиентного бустинга. Устанавливается дерево решений или линейная регрессия на данные, начиная с первого шага для функции F_0 вычисляется погрешность ошибок. Далее формируется новая слабая модель обучающего $h_i(x)$ для отклонений, которая представляет собой функцию разности предыдущего дерева (1), которая является разностью фактического и прогнозируемого результата.

$$h_1(x) = y - F_1(x). \quad (1)$$

На основе двух предыдущих моделей, создаем новую модель F_2, являющейся суммой

$$F_2(x) = F_1(x) + h_1(x), \quad (2)$$

где *x* является объектом обучающей выборки, а *y* ответом обучающей выборки. Функция потерь на каждом шаге обозначается F_i, где *i* меняется от 1 до *M*.

Сравнение алгоритмов прогнозирования. Модель «Lift» сравнивает, насколько хорошо алгоритмы предсказывают выживание пассажиров Титаника по сравнению со случайным угадыванием (рис. 2). Диаграмма говорит о том, что модели на основе дерева (случайный лес, деревья решений с градиентным бустингом) обеспечивают лучший прогноз.

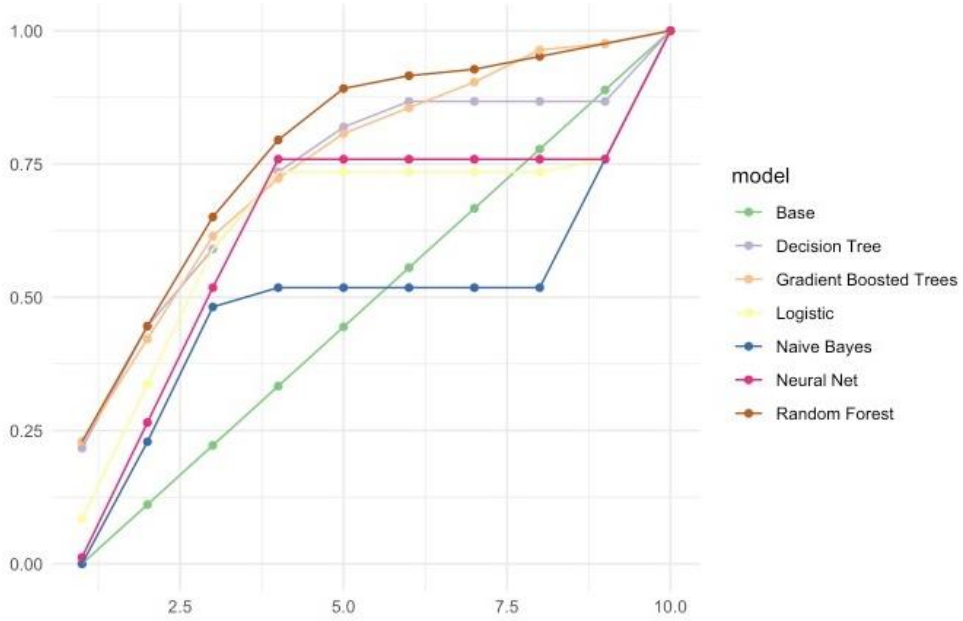


Рис. 2. Модель «Lift» для разных алгоритмов

При обучении был использован датасет «titanic.csv». По оси *y* прогностические параметры (в отн. ед.), *x* элементы выборки сформирование по корзинам.

Заключение. В ходе практики исследованы современные высокоэффективные методологии прогнозирования, и сделан обоснованный выбор технического решения для построения эффективной модели прогнозирования [4]. В ходе работы было осуществлено следующее:

- проведен анализ ансамблевых методов машинного обучения;
- рассмотрен алгоритм градиентного бустинга;
- проведена формализация метода;
- приведен пример пошагового улучшения прогноза;
- проведено сравнение эффективности ансамблевых методов на модели «Lift»;
- рассмотрена зависимость качества прогнозирования от количества деревьев.

Литература

1. Открытый курс машинного обучения. Градиентный бустинг [Электронный ресурс]. – Режим доступа: https://habr.com/en/company/ods/blog/327250/#postanovka-ml-zadachi (дата обращения: 04.01.2019).
2. Градиентый бустинг – просто о сложном [Электронный ресурс]. – Режим доступа: https://neurohive.io/ru/osnovy-data-science/gradientyj-busting/ (дата обращения: 04.01.2019).
3. Gradient Boosting from scratch [Электронный ресурс]. – Режим доступа: https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d (дата обращения: 04.01.2019).
4. Ильясов Э.С. Применение модели градиентного бустинга для прогнозирования развития диабета // Молодой ученый. – 2016. – № 27. – С. 1–5.

Хорошавина Анна Сергеевна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4220
Направление подготовки: 09.04.02 – Автоматизация и управление
в образовательных системах
e-mail: khoroshavina.a@yandex.ru

Русак Алена Викторовна
Год рождения: 1979
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: alena@cde.ifmo.ru

УДК 004.42

ПОДХОД К РАЗРАБОТКЕ МЕТОДА ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ ДЛЯ АВТОМАТИЗИРОВАННОЙ ПРОВЕРКИ ЗНАНИЙ ПО АНГЛИЙСКОМУ ЯЗЫКУ

Хорошавина А.С.
Научный руководитель – к.т.н., доцент Русак А.В.

В работе предложен подход для автоматизации проверки знаний путем распознавания многосимвольного рукописного текста на английском языке, в основе которого лежит сверточная нейронная сеть.
Ключевые слова: распознавание рукописного текста, сверточная нейронная сеть, автоматизация проверки знаний.

Процесс перевода как печатного, так и рукописного текста в электронный текстовый документ называется распознаванием текста. Задача распознавания рукописных символов с изображения (оффлайн распознавание, offline handwriting recognition) довольно сложная, поскольку системе доступна только графическая информация о тексте. Кроме того, существует ряд проблем, которые усложняют процесс распознавания:

– вариативность начертания символов (наклон, размер, соединительные элементы);
– пересечение или наложение текста;
– непараллельность строк;
– орфографические ошибки;
– помарки, исправления или какие-либо дефекты исходного носителя (бумаги);
– качество изображения и прочие.

Однако именно задача оффлайн распознавания представляет интерес для большинства сфер жизни человека, где обработка рукописного текста и преобразование его в электронный формат, помогло бы упростить и автоматизировать долгие рутинные процессы, требующие предельной концентрации внимания со стороны человека. Например, учитель в школе практически ежедневно проверяет сотни письменных работ, что требует напряжения со стороны не только зрительной системы, но и высокой концентрации таких когнитивных функций, как внимание и восприятие.

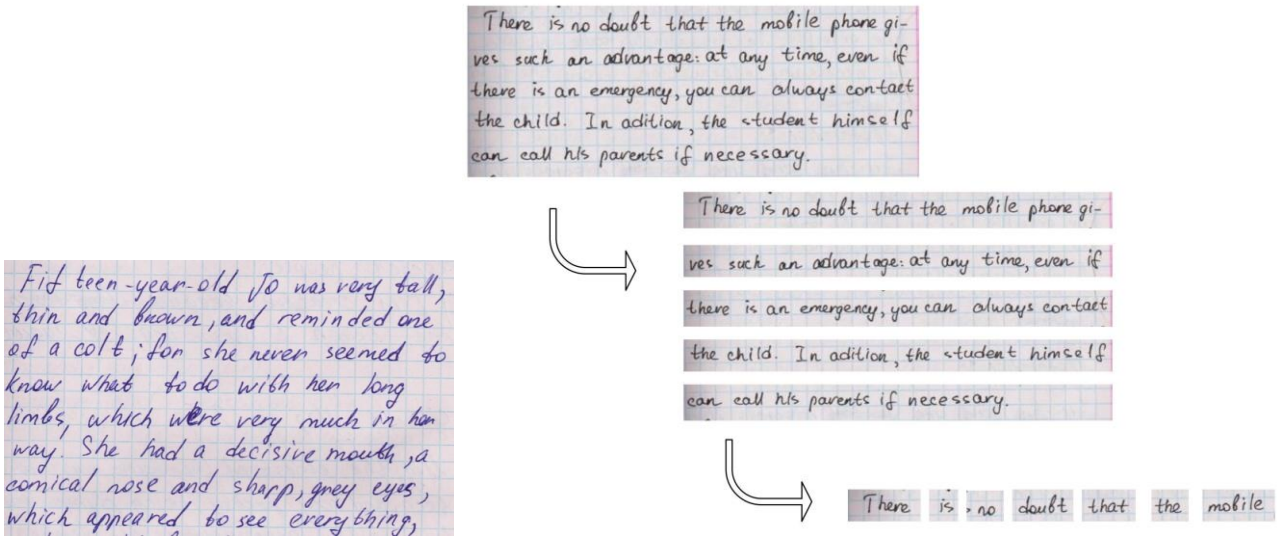
В общем случае зрительное восприятие человека представляет собой процесс распознавания, классификации и идентификации предметов и образов, а распознавание

символов рукописного текста является частным случаем распознавания образов. На сегодняшний момент для решения подобных задач распознавания популярным подходом является использование искусственных нейронных сетей, в частности, сверточных нейронных сетей (СНС, Convolutional Neural Networks, CNN).

Сверточные нейронные сети (СНС) – это особый вид нейронных архитектур, которые были специально разработаны для обработки данных изображения. Благодаря своей архитектуре, такие сети используют пространственную информацию между пикселями изображения и учитывают иерархическую структуру изображения [1, 2].

Ключевыми слоями в архитектуре сверточных нейронных сетей являются сверточный слой (convolutional layer) и слой объединения (pooling/downsampling layer), которые выделяют характерные признаки изображения и уменьшают его размерность, таким образом, делая сеть инвариантной к малым искажениям во входном изображении и уменьшая количество вычислений.

Путем комбинации количества различных слоев и настройки их параметров достигается необходимая точность распознавания. Даже при небольшой глубине (порядка 10 слоев) СНС способна распознать одиночные рукописные символы. Однако задача распознавания текста осложняется тем, что изображение содержит многосимвольную информацию различного характера (буквы, знаки препинания). На рис. 1, а, представлен пример входного изображения для распознавания.



а б

Рис. 1. Изображение, подаваемое на вход системы (а); процесс сегментации изображения (б)

Поэтому для решения подобной задачи необходимо сегментировать изображение на более простые фрагменты (предложения, слова), предварительно определив их местоположение. На рис. 1, б, представлен пример сегментации изображения.

Цель определения местоположения определенных фрагментов текста состоит в размещении вокруг них ограничивающих рамок (bounding box). Широкое распространение получил метод максимально стабильных экстремальных областей (Maximally Stable Extremal Regions, MSER) [3]. Основная идея данного алгоритма – это обнаружение «пятен» на изображении, которые соответствуют тексту.

Однако данный алгоритм не позволяет обнаружить соответствующие фрагменты текста на изображениях подобного формата должным образом, поскольку не имеет достаточно тонкой настройки параметров. В случае мелких символов, таких как знаки препинания, может их проигнорировать, либо произведет слишком мелкую сегментацию, разделив одно слово на несколько отдельных в том случае, когда расстояние между буквами чуть больше, или написание требует наличие дефиса. На рис. 2 представлен пример обнаружения областей текста алгоритмом MSER.

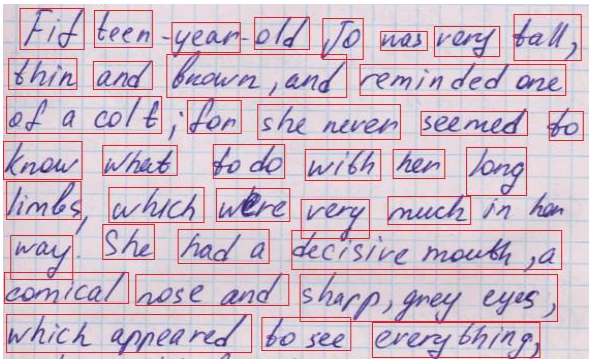


Рис. 2. Результат обнаружения текста алгоритмом MSER

Для решения данной проблемы было решено использовать подход на основе CHC, которая имеет более тонкие настройки и благодаря своей архитектуре справляется с выделением характерных признаков изображения. Для обнаружения текста на изображении и последующей сегментации его на предложения и слова предлагается использовать метод SSD (Single Shot Multibox Detector) [4], в основе которого CHC, и который используется для обнаружения объектов в компьютерном зрении.

Архитектура SSD-метода основана на сверточной сети с прямым проходом, которая создает набор ограничивающих рамок фиксированного размера и оценивает наличие экземпляров классов объектов в этих рамках, после чего для получения окончательных обнаружений применяется алгоритм постобработки, отвечающий за объединение всех обнаружений, принадлежащих одному и тому же объекту. Сначала изображение подается на базовую сеть, используемую для извлечения особенностей изображения, затем добавляется вспомогательная структура для получения обнаружений.

Входными данными для разрабатываемой модели на основе SSD-метода является изображение в градациях серого, которое содержит только рукописный текст. Выходные данные представляют собой ограничивающие рамки, которые соответствуют каждой строке текста, в случае сегментации на предложения, и каждому слову, в случае сегментации на слова. В качестве базовой и вспомогательной сети используются CHC. На рис. 3 представлена модель для обнаружения, сегментации и классификации рукописного текста на основе CHC.

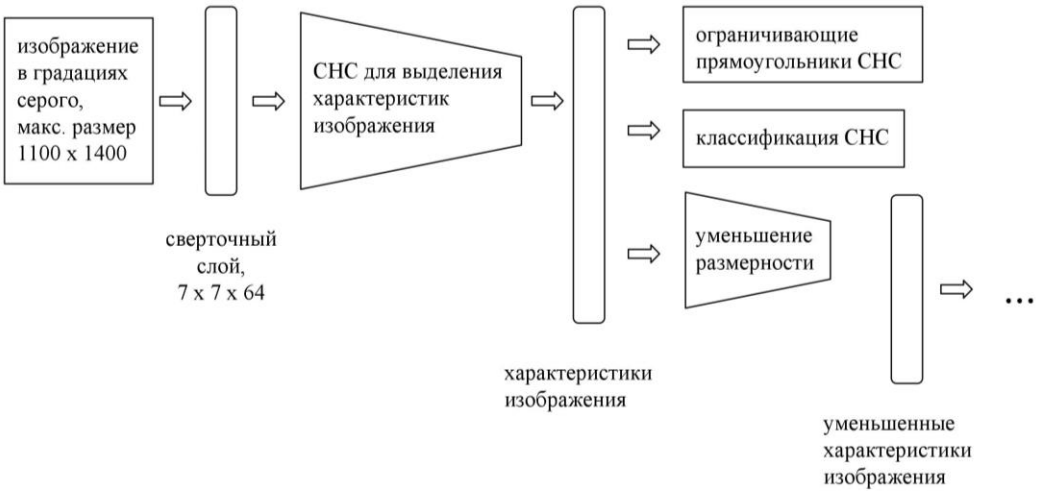


Рис. 3. Модель для распознавания рукописного текста

Таким образом, окончательная модель для автоматизированной проверки знаний будет иметь архитектуру, представленную на рис. 4.

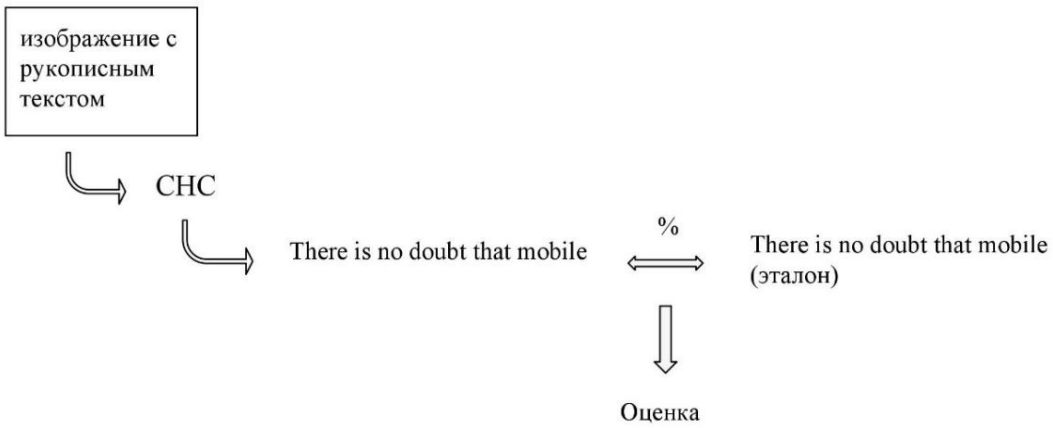


Рис. 4. Схематическое представление системы для автоматизированной проверки знаний

Дальнейшее исследование в рамках данной работы заключается в поиске оптимальных параметров для CHC и выборе программных средств для реализации выбранного метода.

Литература

1. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks [Электронный ресурс]. – Режим доступа: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (дата обращения: 04.01.2019).
2. LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-Based Learning Applied to Document Recognition [Электронный ресурс]. – Режим доступа: http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf (дата обращения: 04.01.2019).
3. Neumann L., Matas J. A method for text localization and recognition in real-world images [Электронный ресурс]. – Режим доступа: http://cmp.felk.cvut.cz/~matas/papers/neumann-text-accv10.pdf (дата обращения: 04.01.2019).
4. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A.C. SSD: Single Shot MultiBox Detector [Электронный ресурс]. – Режим доступа: https://arxiv.org/pdf/1512.02325.pdf (дата обращения: 04.01.2019).

Челокьян Лусинэ Арутюновна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № P4261
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: lusinechelokian@gmail.com

Шалобаев Евгений Васильевич
Год рождения: 1947
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.т.н., доцент
e-mail: shalobaev47@mail.ru

УДК 004.5

СОЗДАНИЕ ИНТЕРФЕЙСОВ ДЛЯ ЛЮДЕЙ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

Челокьян Л.А.
Научный руководитель – к.т.н., доцент Шалобаев Е.В.

В работе определено понятие интерфейс, в частности пользовательский интерфейс, описаны этапы создания интерфейсов. Определены барьеры людей с ограниченными возможностями при работе с интерфейсами, такие как проблемы со зрением или слухом, ограниченная подвижность. Описаны особенности конструирования интерфейсов и управления ими, которые зависят от типа ограничений возможностей пользователя.
Ключевые слова: интерфейс, пользователь, адаптация, люди с ограниченными возможностями, текст.

Согласно оценке глобальной численности населения на 2010 год около 15% населения мира, что составляет более миллиарда человек, живут с какой-либо формой инвалидности [1]. В зависимости от той или иной формы инвалидности, человек может иметь проблемы с повседневными задачами, с которыми человек без ограничений по состоянию здоровья с легкостью справляется. Например, могут возникнуть трудности при взаимодействии с интернет-ресурсами или мобильными приложениями, так как не любой интерфейс адаптирован под человека с ограниченными возможностями.

В широком смысле интерфейс – это определенная некими стандартами граница между взаимодействующими независимыми объектами, или, множество средств и способов, обеспечивающее взаимодействие между двумя структурами или системами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов. Из определения следует, что практически все сталкиваются с интерфейсами, вне зависимости от использования мобильных устройств или интернет-ресурсов [2]. Пользовательский интерфейс (User Interface, UI) – интерфейс, который обеспечивает передачу информации между человеком (пользователем) и программно-аппаратными компонентами компьютерной системы.

Этапы разработки пользовательского интерфейса:

1. проектирование: определение цели, постановка задачи, определение требований пользователя, оценка взаимодействия с пользователем, моделирование;
2. реализация: создание прототипа, макета предмета;

3. тестирование: выявление несоответствий на 1 этапе проектирования и прототипа в процессе тестирования.

Непосредственно проектирование интерфейса включает в себя несколько этапов от исследований до предоставления материалов веб-разработчику (рис. 1). На этапе исследования проводится сбор информации о продукте, его аналогах, потребителях, анализ устройств предполагаемой целевой аудитории. Этот этап определяет ограничения и возможности разработки. На основе предоставленного описания работы интерфейса создается список задач (пользовательских сценариев), которые может выполнять пользователь в рамках интерфейса. Пошаговое описание задач позволяет определить оптимальную длину решения задачи. Полученный список шагов составляет основу структуры интерфейса, определяет количество экранов и их содержание. Черновой прототип наглядно демонстрирует структуру, объем и варианты развития сценариев. Финальный прототип включает в себя контент, кнопки и прочие элементы. После этапа исследования и параллельно с этапами проектирования идет определение будущей стилистики интерфейса. Выбор стилистики также зависит от целевой аудитории и содержания. Дизайн-концепция демонстрирует будущий вид интерфейса и то, как выбранная стилистика соответствует содержанию. Оформление всех экранов производится после утверждения дизайн-концепции. Принимается окончательное решение относительно всех элементов интерфейса, от интерлиньяжа до иконок и изображений. Анимация интерфейса демонстрирует то, как он будет функционировать. Также анимация необходима разработчикам для понимания замысла дизайнера. Завершающим этапом является передача систематизированных материалов разработчикам [3].

Рис. 1. Этапы разработки интерфейсов

При конструировании интерфейса разработчик обычно знает, для какой категории пользователей данный интерфейс предназначается. Чтобы учесть возможность использования приложения с таким интерфейсом другими категориями пользователей, необходимо составить онтологическую модель пользователя, содержащую в себе необходимые свойства каждого типа пользователя для выбора наилучшего способа адаптации интерфейса. Классификацию

пользователей можно произвести по их возможностям: физическим, умственным, и способности к восприятию окружающего мира. Также для адаптации интерфейса немаловажным фактором является устройство, которым пользуется человек с ограниченными возможностями, а именно, его характеристики с точки зрения аппаратной и программной частей. Зная информацию о возможностях пользователя и о свойствах его девайса, можно выбрать подходящие формы адаптации интерфейса [4].

Людей с ограниченными возможностями можно разделить на следующие группы по типу ограничений:

– слепота или плохое зрение;
– цветовая слепота;
– ограниченная подвижность и двигательная неспособность;
– глухота или расстройства слуха;
– когнитивные расстройства.

Каждая группа имеет свои типичные потребности. При разработке интерфейсов для людей с плохим зрением и слепотой следует продумать управление элементами с помощью сочетаний клавиш, разработать звуковые оповещения и пояснения визуальных объектов, а также использовать крупный шрифт. При этом важно грамотно его подобрать, необходимо учитывать особенности восприятия отдельных букв со сложной структурой (рис. 2), текст должен быть черно-белым или контрастным.

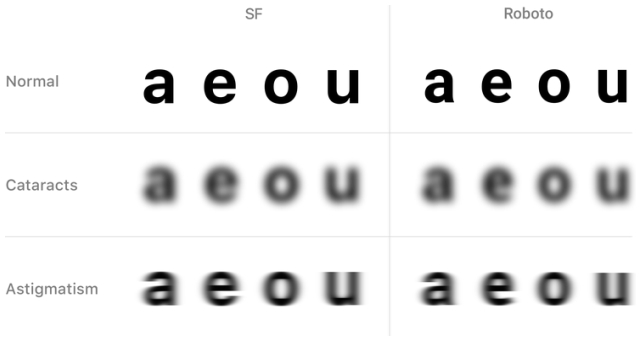


Рис. 2. Пример восприятия шрифтов Roboto и San Francisco людьми с различными расстройствами зрения [5]

Цветовая слепота может быть трех типов: протанопия, дейтеранопия, тританопия. В зависимости от типа цветовой слепоты необходимо избегать определенных цветовых сочетаний или дублировать отображение информации другими средствами. Стоит использовать яркие цвета и цветовые схемы с высокой контрастностью. Например, черно-белые схемы помогают пользователям, страдающим цветовой слепотой, различать цвета. Текст необходимо размещать на простом фоне без узоров. Не рекомендуется использовать цвет в качестве единственного визуального средства передачи информации или обозначения действия. На рис. 3 изображена стандартная палитра цветов для людей с различными отклонениями в восприятии цвета. Именно поэтому, в процессе дизайна стоит комбинировать цвет с другими идентификационными знаками, например, пояснительным текстом [6].

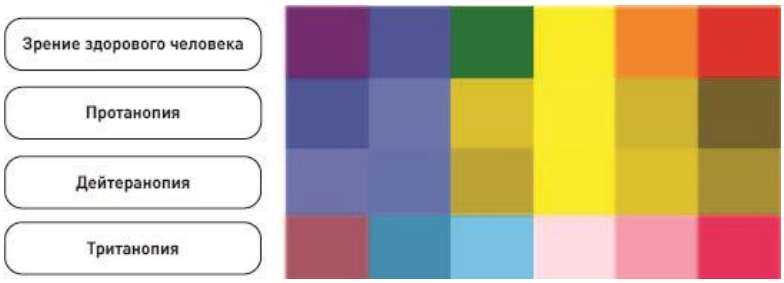
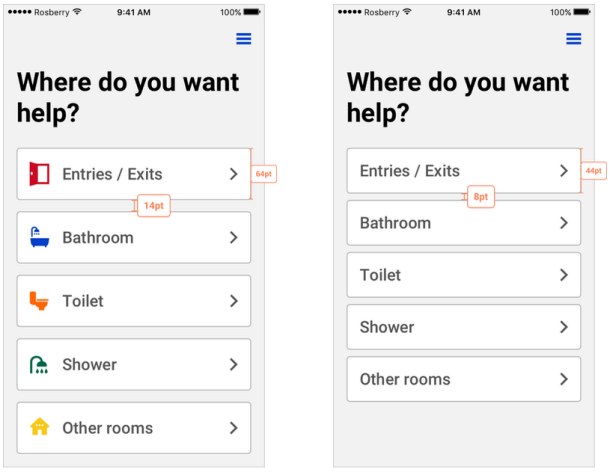


Рис. 3. Палитра восприятия цветов людьми с различными расстройствами [6]

Людям с ограниченной подвижностью и двигательной неспособностью необходимо предоставить возможности для пропуска длинных списков и отмены основных действий, которые позволят пользователю отменить нежелательные действия, ссылки должны быть с полным текстом, а кнопки крупными, чтобы на них было проще «попасть».

Для людей с расстройствами слуха или глухотой необходима текстовая расшифровка звука и видео, субтитры, понятные описания, в которых приведены альтернативы указаниям прослушать что-либо.

Людям, страдающим дислексией, кажется, что текст плавает на странице; текст сливается или искажается. Их потребности аналогичны требованиям пользователей с ослабленным зрением, и им важно, чтобы текст было максимально удобно читать. Пользователям с дефицитом внимания сложно сконцентрироваться на предложении, задаче и т.д. Им требуется простое оформление сайта, которое помогает сосредоточиться. Кнопки должны быть достаточно большими, чтобы по ним было легко «попасть», что особо актуально для пользователей с нарушениями моторики и дислексией. На рис. 4, а, показан грамотный пример подбора размера кнопок и расстояния между ними, рис. 4, б – неудачный пример [5].



а б

Рис. 4. Пример подбора размера кнопок [5]: грамотный (а) и неудачный (б)

На основании вышеизложенного, следует отметить, что необходимо учитывать особенности людей с ограниченными возможностями при проектировании интерфейсов для них. В данной работе выявлены основные этапы создания интерфейсов, определены типы ограничений и описаны основные требования к разработке интерфейсов для людей с ограниченными возможностями.

Литература

1. Hartley S.D. World report on disability. – World Health Organization, 2011. – 350 p.
2. Cooper A. The Essentials of Interaction Design. – LA, 2010. – 688 p.
3. Krug S. Don`t make me think! A common sense approach to web usability. – New Riders Publishing Berkeley: California USA, 2008. – 224 p.
4. Дворянкин А.М., Романенко Р.Р., Поцелуйко А.С. Разработка алгоритма адаптации интерфейсов для людей с ограниченными возможностями // Изв. Волгоградского государственного технического университета. – 2015. – С. 49–55.
5. [Электронный ресурс]. – Режим доступа: https://apptractor.ru/info/articles/diy-kak-myi-razrabatyivali-mobilnoe-prilozhenie-dlya-lyudey-s-ogranichennyimi-vozmozhnostyami.html (дата обращения: 04.01.2019).
6. [Электронный ресурс]. – Режим доступа: http://www.cmsmagazine.ru/library/items/graphical_design/nesting-sites-for-people-with-disabilities/ (дата обращения: 04.01.2019).

Шайхетдинова Алиса Раисовна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4263
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: alisa.shayk@gmail.com

Шуклин Дмитрий Анатольевич
Год рождения: 1979
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: dmshuklin@niuitmo.ru

УДК 004

ИССЛЕДОВАНИЕ МЕТОДОВ ПОИСКОВОГО МАРКЕТИНГА В МОБИЛЬНОЙ ВЫДАЧЕ

Шайхетдинова А.Р.
Научный руководитель – к.педагог.н., доцент Шуклин Д.А.

В работе выделены методы продвижения в органической и контекстной выдаче с учетом тенденции мобильного поиска. Также подготовлена база для проведения исследования поведения пользователей при мобильном поиске в поисковых системах (показатели: возраст, вовлеченность, запрос, канал). Определены этапы планирования, реализации и отчетности на основе модели EVM.
Ключевые слова: поисковый маркетинг, интернет-продвижение, SEO, SEA, мобильный поиск.

Поисковый маркетинг – комплекс мер, направленный на увеличения целевого трафика на сайт посредством обеспечения его видимости в выдаче поисковых машин. Существуют два главных источника трафика в поисковых системах (ПС): органический SERP (Search Engine Results Page) и контекстная реклама [1].

Для оптимизации сайта под органический поиск необходимо уделить внимание четырем категориям: техническая сторона, контент, ссылочный профиль и поведенческие факторы. Техническая оптимизация направлена на то, чтобы обеспечить поисковому роботу простой проход и индексацию страниц сайта. Работа над контентом включает в себя сбор семантического ядра, начиная от масок и базисов, заканчивая long-tail запросами. Кроме того, этот сегмент SEO (Search Engine Optimisation) ориентирован на создание оптимальной структуры и оптимизированного контента с целью более высокого ранжирования. Ссылочный профиль подразумевает под собой значимость страницы в сети. Чем больше качественных ссылок ведет на сайт, тем вес ее больше. И, наконец, поведенческий аспект включает в себя заботу о продолжительности сессий и показателе отказов, которые необходимо оптимизировать посредством, например, UX-дизайна.

Эффективная стратегия оптимизации рекламных кампаний для платной выдачи ПС состоит из нескольких последовательных действий [2]:

1. сбор ключей для продвижения и определение релевантных посадочных страниц для них;
2. написание оптимизированных текстов и расширений для богатого сниппета и высокого CTR (click-through rate);
3. настройка расписания и геолокации для кампаний;

4. постоянная аналитика ROI (Return On Investment);
5. оптимизация ставок для аукциона на основе показателей аналитики.

Стратегии оптимизации сайта и рекламных кампаний, описанные выше, являются основными. Однако ввиду увеличения доли мобильного трафика растет необходимость оптимизации и под мобильные устройства [3], ведь в современной выдаче попасть в видимый топ выдачи стало сложнее из-за размера экрана. Поэтому выход в контекстный блок и расширение сниппета становятся критически важными.

В SEO добавляется above-the-fold оптимизация, суть которой состоит в том, чтобы не подгружать скрипты при загрузке первого экрана сайта; использование технологии AMP (Accelerated Mobile Pages) также актуально. QA и how-to разделы являются методами оптимизации под голосовой поиск. Для продвижения по органике критично использование Google My Business и микроразметки Schema.org, для облегчения работы индексирующим роботам и расширения сниппета.

В SEA для настройки кампании для мобильного пользователя заполняется чекбокс в Яндекс.Директ. В Google.Ads разделяются кампании и корректируются ставки. При создании расширенных рекламных объявлений не допускается использование более 4-х быстрых ссылок, потому что иначе появляется горизонтальная прокрутка, негативно влияющая на usability. Кроме того, компании, допускающие прием «холодных» звонков, могут добавить телефонный номер в виртуальную визитку, тогда пользователи могут переходить к звонку непосредственно из SERP. В таком случае необходимо использовать систему статического колтрекинга для аналитики рекламной кампании.

В силу постоянной конкуренции в выдаче ПС, маркетологи и оптимизаторы находятся в непрерывном поиске новых инсайтов для улучшения бизнес-показателей. Анализ пользовательского поведения в ПС может выявить зависимости между характеристиками пользователей, их запросами и выбором канала маркетинга [4].

Показатели: возраст, вовлеченность, запрос, канал.

Цель: определить, какой канал поискового маркетинга является наиболее подходящим для компаний различной направленности, organic или PPC, в мобильной выдаче. При обнаружении зависимостей между показателями можно сформулировать ранее неизвестные рекомендации для планирования стратегии поискового маркетинга.

Вопросы дальнейшего исследования:

– как распределены клики в SERP при поиске товаров низкой и высокой вовлеченности;
– что выбирают в поисковой выдаче представители различных поколений;
– ключевые фразы какого формата используются для поиска товаров низкой и высокой вовлеченности;
– какие ключевые фразы используются поколениями X и Y при поиске различных товаров.

Для проведения эксперимента планируется использование метода виньеток (EVM) [5]. Для каждого сценария и испытуемого будет записана ключевая фраза, страница SERP и канал перехода. Для анализа данных будет использован обычный дисперсионный анализ (ANOVA). Однако для использования этого метода необходимо иметь унифицированные показатели, для этого были определены категории для ключевых слов:

1. маски и long-tail;
2. продуктовые и брендовые.

Ценность настоящей работы заключалась в детальном обзоре методов продвижения ресурса в органической и контекстной мобильной выдаче и в организации плана проведения исследования пользовательского поведения при поиске на мобильном устройстве. В будущем данная работа будет продолжена с целью практического проведения исследования пользователей и анализа полученных данных. На основе полученных результатов будут предложены рекомендации по планированию стратегий поискового маркетинга.

Литература

1. Неелова Н. SEMbook Энциклопедия поискового продвижения Ingate. – Изд. 2-е. – ИП Андросов, 2017. – 541 с.
2. Joshi A., Motwani R. Keyword Generation for Search Engine Advertising // 6th IEEE International Conference on Data Mining. – 2006. – P. 490–498.
3. Qian F. Toward Mobile-Friendly Web Browsing // IEEE Internet Computing. – 2015. – V. 19(5). – P. 72–76.
4. Zhitomirsky-Geffet M., Blau M. Cross-generational analysis of information seeking behavior of smartphone users // Aslib Journal of Information Management. – 2017. – V. 69(6). – P. 721–739.
5. Aguinis H., Bradley K.J. Best practice recommendations for designing and implementing experimental vignette methodology studies // Organizational Research Methods. – 2014. – V. 17(4). – P. 351–371.

Шаповалова Дарья Викторовна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р41601
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: shapovalovadaria@yandex.ru

Готская Ирина Борисовна
Год рождения: 1957
Университет ИТМО, факультет программной инженерии и компьютерной техники, д.педагог.н., профессор
e-mail: iringot@mail.ru

УДК 004.4'275

АНАЛИЗ НАУЧНЫХ ИССЛЕДОВАНИЙ ПО ПРОБЛЕМЕ СОЗДАНИЯ 2D- И 3D-АНИМАЦИИ В МУЛЬТИПЛИКАЦИИ

Шаповалова Д.В.
Научный руководитель – д.педагог.н., профессор Готская И.Б.

В работе уточнены основные термины и понятия, имеющие неоднозначные трактовки и необходимые для проведения исследования, а также рассмотрены программы для создания 2D- и 3D-анимации в мультипликации, и представлены разработанные критерии для проведения их сравнительного анализа.
Ключевые слова: 2D- и 3D-анимация, программы для создания анимации, мультипликация, сравнительный анализ программ.

Есть множество программ для создания анимации, но далеко не все из них подходят для мультипликации. В свою очередь, среди тех программ, которые используются для мультипликации, не все подходят для создания полноценной анимации, например, отдельные программные средства (ПС) подходят только для создания персонажей, а для создания собственно мультфильма необходимо использовать другие ПС. Вследствие этого очевидна проблема, какие из ПС можно использовать для анимации в мультипликации. Анализ научных исследований и рекомендаций разработчиков, размещенных в сети Интернет, показал отсутствие систематизированного списка ПС для создания анимации, построенного на основе результатов их сравнительного анализа. Все вышесказанное определяет актуальность проводимого исследования.

Для однозначного понимания используемых в исследовании терминов и понятий необходимо уточнить те из них, которые имеют многозначное трактование в словарях, научных исследованиях и методических разработках.

«Мультфильм – мультипликационный фильм» [1]. Основным понятием, используемым в исследовании, является «мультипликация», под которой понимается:

– «съемка отдельных рисунков или объемных фигур, изображающих последовательные фазы движения, что создает на экране иллюзию движения неподвижных объектов, а также сами рисунки, фигуры, являющиеся объектом такой съемки;
– фильм, сделанный посредством такой съемки» [1].

В данном исследовании понятие «мультипликация» используется в значении мультипликационный фильм.

В большинстве работ выделяют шесть видов мультипликации: кукольная, песочная, пластилиновая, рисованная, ротоскопирование и компьютерная анимация. Целью работы являлся анализ компьютерной анимации, остальные выходят за рамки сферы IT.

Следующее понятие, которое требует уточнения – «анимация», используемое в некоторых работах как синоним мультипликации. В частности, толковый словарь Ефремовой [2] определяет это понятие как:

– «киносъемка последовательных фаз движения рисованных или объемных фигур, в результате которой при показе на экране у зрителя создается иллюзия их движения; мультипликация;
– объект такой съемки».

Компьютерная анимация – анимация, создаваемая на компьютере, бывает двух видов: 2D- и 3D-анимация.

Создание компьютерной анимации (2D и 3D) можно разделить по следующим видам:

1. покадровая анимация (по ключевым кадрам);
2. процедурная анимация;
3. шейповая анимация (shape-анимация);
4. стоп-моушен (stop motion);
5. запись движения (метод захвата движения);
6. синтез движения.

Многие из данных видов анимации существовали и ранее, только выполнялись вручную; в настоящее время с появлением различных компьютерных программ создание анимации отчасти упростилось.

– Покадровая анимация (frame by frame) (анимация по ключевым кадрам) представляет собой последовательность ключевых кадров. Они служат для внесения изменений в анимацию, создаются разработчиком; промежуточные кадры могут быть созданы автоматически. Используется в 2D- и 3D-анимации.
– Процедурная анимация – анимация, используемая для воссоздания таких эффектов и движений, которые непросто воспроизвести с помощью ключевых кадров. Рассчитывается благодаря неким начальным значениям математических выражений, которые описывают изменение параметров во времени. Используется в только в 3D-анимации.
– Шейповая анимация или shape анимация (от английского «shape» – форма) – анимация, в которой преобразовываются простейшие геометрические фигуры. Используется в 2D-анимации.
– Стоп-моушен (stop motion) – это анимация, основанная на методах кукольной и пластилиновой мультипликации. Используется в 3D-анимации.
– Запись движения (метод захвата движения) – анимация, основанная на наложении движений реальных объектов на виртуальные модели. На реальные объекты устанавливаются специальные датчики, которые отслеживают движения. Потом эти движения оцифровываются и записываются в определенном формате. Используется только в 3D-анимации.
– Синтез движения – анимация, в которой могут выполняться следующие задачи: комбинирование движений (несколько движений комбинируется в одно) и заимствованные движения (спроектированное для одного объекта может переноситься на другой объект, схожий по структуре). Используется в 3D-анимации.

Можно также к вышеописанным видам анимации добавить еще следующие:

– программируемая анимация – анимация, созданная на одном из языков программирования, изменяющая в течение определенного промежутка времени некое количество параметров. В таком виде можно записать как 2D-, так и 3D-анимацию;
– еще один метод анимации, который применяется как в 3D-анимации, так и в 2D-анимации. Скелетная анимация – это такой метод анимации, при котором использована иерархичная структура костей. Каждая кость занимает свое место в иерархии скелета и подвергается

влиянию других костей; каждая дочерняя кость наследует трансформацию родительской кости. Бывает с прямой и инверсной кинематикой.

ПС, используемые для создания анимации, в основном англоязычные. Их можно приобрести как платно, так и бесплатно на официальных сайтах компаний-разработчиков, а также их посредников. Одни программы достаточно просты в освоении и использовании, другие требуют сформированных навыков по работе с анимационными ПС.

Программы для 2D-анимации

1. Если говорить о компании, которую можно назвать лидером в области создания программ для анимации, то такой компанией является канадская компания Toon Boom Animation. Но в данной работе речь пойдет только об одной программе этой компании, наиболее известной как Toon Boom Harmony (раннее Toon Boom Opus). В дополнении к созданию анимации в этой программе можно осуществить сетевую работу нескольких пользователей, новый инструмент «true pencil», работу с частицами и другие дополнения. Отсутствует русский язык.
2. Moho Pro 12 (ранее Anime Studio Pro) – программное обеспечение (ПО) компании Smith Micro Software. Создает векторную 2D-анимации и графику. Подходит для создания профессиональной анимации. Отсутствует русский язык.
3. Adobe Animate (ранее Adobe Flash Professional, Macromedia Flash и Future Splash Animator) – программа, разработанная Adobe Systems. Используется для создания интерактивных векторных и растровых анимаций для: игр, мультипликации, приложений и интернета. В создании мультипликации используется нечасто.
4. Easy GIF Animator – это мощный анимированный GIF-редактор и топ-инструмент для создания анимированных изображений, баннеров, кнопки и GIF-видео. Для мультипликации используется редко, больше предназначен для коротких роликов.
5. TVPaint Animation – (также известный как TVPaint, TVP, Bauhaus Mirage или NewTek Aura) – это ПО для создания 2D-анимации. Было разработано TVPaint Developpement SARL, основанная в Лотарингии (Франция). Отсутствует на русском языке.
6. Animation Paper (ранее был Plastic Animation Paper) – программа для создания 2D-анимации. На данный момент считается устаревшей, создание анимации близко к созданию рисованной мультипликации. Отсутствует на русском языке.
7. Pencil2D Animation (ранее Pensil) – развивающееся ПО, которое специализируется на 2D-анимации. Поддерживает работу с растровой и векторной графикой, есть классический набор функций и инструментов для создания двумерной анимации.
8. Synfig Studio – это бесплатное ПО с открытым исходным кодом, лицензированное под GNU GPL v2. Только 2D-анимация.

Программы для 3D-анимации

1. «Autodesk Maya – программа для 3D-анимации, моделирования и визуализации, предоставляющая мощный интегрированный инструментарий, который можно использовать для создания анимации, сред, графики движения, виртуальной реальности и персонажей» [из официального сайта].
2. iClone (ранее iClone Pro) – самое быстрое и самое рентабельное ПО для 3D в индустрии, помогающее вам легко произвести профессиональные анимации за очень короткое время для фильмов, превизов, анимации, видеоигр, образования и искусства.
3. MODO – ПС для 3D-моделирования, анимации и текстуирования, созданная Luxology, LLC.
4. Blender – профессиональное открытое ПО для создания 3D-графики. Он поддерживает все средства работы с компьютерной графикой – моделирование, рендеринг, анимацию, добавление звука в видео, постобработку, компоновки с помощью Node Compositing и создание интерактивных игр.

5. Dragonframe – это ПО для стоп-моушн анимации. В этой программе были сделаны такие мультфильмы, как «Коралина в Стране Кошмаров», «Франкенвини» и т.д.
6. Autodesk Softimage, или просто Softimage (ранее также Softimage|XSI) – многофункциональный редактор 3D-графики, сейчас принадлежит Autodesk. Подходит для 3D-моделирования, создания спецэффектов и анимации. Используется в рекламе, в кино, для видеоигр. Может подойти и для мультипликации.

Критерии оценивания ПС для создания 2D- и 3D-анимации

– Качество получаемой анимации. Нет сомнений, чем лучше качество получившейся анимации, тем значимее программа для создания анимации. В мультипликации особенно важно, чтобы полученное изображение не было размытым, оказалось в меру ярким, в меру контрастным и т.д. К сожалению, во многом понятие качества является субъективным, и потому точных оценок измерения качества не может быть.
– Доступность и платность ПС. Не все программы для создания анимации можно найти в свободном доступе, особенно новые их версии. Для одних программ достаточно лишь зарегистрироваться на сайте, за другие еще придется и доплатить, а есть такие, для которых существуют специальные лицензии и договора. Также необходимо отметить условия, на которых предоставляется использование, как и самого продукта, так и отдельных его компонентов.
– Цена (для платного ПО). Цену тоже нужно учитывать, подбирая ПО. Есть очень дорогие программы, а есть программы и по приемлемой цене. Если основной статьей расхода будет являться стоимость ПО, то нужно ли покупать именно это дорогое ПО? Многие платные программы возможно оценить с помощью демоверсии, предоставляемой на определенный период времени и с ограниченным функционалом.
– Поддержка и новые версии. При выборе программ для создания анимации важно учитывать, насколько часто в данных программах встречаются ошибки, как часто их исправляют и насколько часто выходят новые версии данного программного продукта. Наличие ошибок необходимо учитывать и при переходе на новую версию, так в некоторых случаях новая версия может работать хуже предыдущей.
– Совместимость программ с конкретной операционной системы (ОС). Если у ПО отсутствует совместимость с ОС, то программа не сможет работать на данном компьютере, а настройка новой системы слишком дорого обойдется.
– Объем потребляемых ресурсов, памяти на компьютере и другие системные требования. Быстродействие программы зависит от того, насколько эффективно они потребляют ресурсы. К сожалению, ПС для создания анимации могут потреблять очень много ресурсов и памяти, однако, по этому критерию ПС оценивают далеко не в первую очередь. При этом очень «тяжелые» программы могут сильно замедлить работу компьютера или даже нанести ему вред, поэтому объем потребляемых ресурсов и памяти следует обязательно учитывать.
– Вид анимации. Выбор нужной программы зависит от того, какую графику нужно использовать в анимации: растровую, векторную, фрактальную или 3D. Разные программы работают с разными видами графики и используют разные средства для создания анимации.
– Функционал. Разные программы отличаются как способами создания анимации, так и наличием различных эффектов: различаются подходы к созданию и изменению моделей, возможности создания симметричных изображений, динамики мягких и твердых тел и т.д. Для решения определенных задач важно учитывать функционал, чтобы иметь возможность выполнить эти задачи.
– Понятность интерфейса. У разных программ разный функционал и, соответственно, разные способы реализации этого функционала. Интерфейс (в том числе и англоязычный или на любом другом иностранном языке) должен быть интуитивно понятен пользователю, т.е. использование различных функций не должно вызывать затруднений. К сожалению, чем более разнообразный функционал, тем труднее становится интерфейс, и программы для

создания анимации не являются исключением из этого правила. Поэтому ПС надо выбирать, руководствуясь не только вышеприведенными функциями, но и с точки зрения удобство использования программ, возможности сразу найти искомую функцию.

– Создаваемые форматы. Не все создаваемые файлы анимации можно просматривать на любом носителе, поэтому при выборе программы также важно учесть форматы тех видеофайлов, которые предлагает программа для создания анимации [3–6].

Выделенные критерии в дальнейшем планируется использовать для проведения сравнительного анализа ПС и обоснованного выбора наиболее эффективного ПС для создания анимации.

Литература

1. Малый академический словарь [Электронный ресурс]. – Режим доступа: https://gufo.me/dict/mas (дата обращения: 04.01.2019).
2. Ефремова Т.Ф. Современный толковый словарь русского языка: в 3 т. – М.: АСТ, Астрель, Харвест.
3. Словарь русского языка: в 4-х т. / Под ред. А.П. Евгеньевой. – 4-е изд., стер. – М.: Рус. яз.; Полиграфресурсы, 1999.
4. Голубев В.О., Кацуба В.С. Мультимедийная презентация учебной информации: практические аспекты реализации [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/v/multimediynaya-prezentatsiya-uchebnoy-informatsii-prakticheskie-aspekty-realizatsii (дата обращения: 04.01.2019).
5. Maya / Программа для компьютерной анимации и модулирования / Autodesk [Электронный ресурс]. – Режим доступа: https://www.autodesk.ru/products/maya/overview (дата обращения: 04.01.2019).
6. blender.org / Home of the Blender project / Free and Open 3d Cretion Sofrware [Электронный ресурс]. – Режим доступа: https://www.blender.org (дата обращения: 04.01.2019).

Швалева Дарья Сергеевна
Год рождения: 1995
Университет ИТМО, факультет программной инженерии и компьютерной техники, студент группы № Р4262
Направление подготовки: 09.04.02 – Информационные системы и технологии
e-mail: dashka.shvaleva@mail.ru

Государев Илья Борисович
Университет ИТМО, факультет программной инженерии и компьютерной техники, к.педагог.н., доцент
e-mail: goss@corp.ifmo.ru

УДК 004.4

КРИТЕРИИ СРАВНЕНИЯ МЕТОДОЛОГИЙ УПРАВЛЕНИЯ ОБЪЕКТНОЙ МОДЕЛЬЮ ВЕБ-ДОКУМЕНТА

Швалева Д.С., Государев И.Б.
Научный руководитель – к.педагог.н., доцент Государев И.Б.

Работа выполнена в рамках темы НИР № 615892 «Исследование и разработка в области информационных технологий».

В работе рассмотрены методологии управления объектной моделью веб-документа, как jQuery, React.js и JavaScript. Выделены и сравнены общие характеристики и возможности методологий, на основе которых выявлены общие критерии для практического сравнения.
Ключевые слова: методология, объектная модель документа, DOM, JavaScript, критерии.

Каждый год с развитием технологий и увеличением количества пользователей глобальной сети возрастает веб-нагрузка, что приводит к возрастанию требований к производительности веб-сайтов и веб-приложений. Для разработчиков становится важна скорость обработки процессов, скорость написания продукта, что отчасти зависит от объема написанного кода [1–5].

Разработчики имеют возможность изменять интерфейс и взаимодействовать с интерфейсом пользователя через DOM, что дает им огромную мощь и гибкость при программировании.

jQuery – это одна из таких библиотек, которая значительно упрощает и ускоряет написание JavaScript кода. Девиз jQuery – «write less, do more», что означает – «пиши меньше, делай больше», отражает ее главное предназначение.

Библиотека поразила всех своей простотой, изящностью, магией. Разработчики работают с jQuery с удовольствием, пишут плагины для jQuery, даже пишут статьи про jQuery. Она однозначно стала стандартом в веб-разработке.

Библиотека работает со всеми браузерами. Данная особенность дает возможность больше не беспокоиться о кросс-браузерной совместимости JavaScript кода.

Разработчики выбирают jQuery по следующим причинам:

1. jQuery позволяет добиться большего результата при меньшем объеме кода;
2. методы jQuery применимы к целым группам элементов;

3. библиотека jQuery справляется с различиями в реализации DOM в различных браузерах.

Библиотека jQuery имеет открытый исходный код. Разработчик, в случае если результат не совпадает с ожидаемым, может обратиться непосредственно к коду библиотеки на JavaScript и, если это необходимо, внести соответствующие изменения.

React.js – это библиотека JavaScript, созданная разработчиками Facebook и Instagram, которая используется для создания пользовательского интерфейса.

Для решения проблемы производительности была разработана концепция виртуального DOM, и отличительной особенностью React.js является то, что данная библиотека работает именно с Virtual DOM, а не обычным.

Если приложению нужно узнать информацию о состоянии элементов, то происходит обращение к Virtual DOM. Если необходимо изменить элементы веб-страницы, то изменения вначале вносятся в Virtual DOM. С Virtual DOM разработчики могут добиваться высокой производительности приложений.

В итоге такая схема взаимодействия с элементами веб-страницы работает гораздо быстрее и эффективнее, чем, если бы работать из JavaScript с DOM напрямую.

Основными преимуществами при выборе React.js для разработчика станут:

1. каждый компонент будет отрисован, глядя на исходных код;
2. Virtual DOM может повысить производительность высоконагруженных приложений, что может снизить вероятность возникновения возможных неудобств и улучшить пользовательский опыт;
3. использование изоморфного подхода помогает производить рендеринг страниц быстрее, что помогает сократить ожидание пользователя при загрузке страницы или приложения. Поисковые системы индексируют такие страницы лучше. Поскольку один и тот же код может быть использован как в клиентской, так и в серверной части приложения, нет необходимости в дублировании одного и того же функционала. В результате время разработки и затраты снижаются;
4. переиспользование кода позволяет проще создавать мобильные приложения. Код, написанный во время создания сайта, может быть использован для создания мобильного приложения. Если в планы разработчика входит не только создание сайта, но и мобильного приложения, тогда разработчику нет необходимости искать людей для создания мобильного приложения.

Document Object Model не является языком программирования, но без JavaScript он не имел бы никакой модели или представления о веб-странице, HTML-документе, XML-документе и их элементах.

JavaScript – язык сценариев, или скриптов. Скрипт представляет собой программный код – набор инструкций, который не требует предварительной обработки.

Разработчики при написании веб-приложений часто используют JavaScript из-за некоторых ключевых особенностей языка, а именно:

– приведение типов данных проводится автоматически;
– функции выступают объектами базового класса;
– автоматическая очистка памяти. Так называемая сборка мусора делает JavaScript похожим на C# или Java.

Популярность JavaScript открывает перед программистом немалое количество готовых библиотек, которые позволяют значительно упростить написание кода и нивелировать несовершенства синтаксиса. Однако написание кода на чистом JavaScript является долгой и тягостной работой для программиста.

Программируя с JavaScript нужно писать свои собственные сценарии, которые отнимают достаточно много времени. Так как у него нет готовых написанных сценариев. А также с помощью JavaScript необходимо обрабатывать совместимость с несколькими браузерами, написав собственный код для каждого.

По сравнению с библиотеками, которые были рассмотрены выше, разработчику JavaScript не нужно импортировать и добавлять библиотеки для запуска кода. Код просто должен быть записан внутри тега <script> и </ script> в HTML или подключен отдельным файлом.

Исследовав разные источники и литературу для дальнейшего сравнения были выбраны следующие критерии: тип, язык, простота использования, совместимость, длина кода, скорость, клиентский/серверный язык, объем файлов, способ подключения, синтаксис.

Таблица. Сравнение jQuery, React.js и JavaScript

Критерий	jQuery	React.js	JavaScript
Тип	Библиотека JavaScript	Библиотека JavaScript	Язык программирования
Язык	Использует ресурсы, предоставляемые JavaScript, чтобы упростить процесс	Использует ресурсы, предоставляемые JavaScript, чтобы упростить процесс	Написан на C. Это интерпретируемый язык
Простота использования	Не нужно много писать, так как сценарии уже существуют	Не нужно много писать, так как сценарии уже существуют	Необходимо писать свои собственные сценарии, которые отнимают много времени
Совместимость	Кроссплатформенный	Кроссплатформенный	Необходимо обрабатывать совместимость с несколькими браузерами, написав собственный код
Длина кода	Благодаря сценариям сокращается написание кода	Благодаря сценариям сокращается написание кода	Код длинный и громоздкий
Скорость обработки	Быстро работает с современными браузерами и компьютерами. Подходит для сложных операций	Работает с Virtual DOM, что сокращает время обработки, увеличивая скорость обработки	Чистый JavaScript быстрее получает доступ к DOM, так как сокращает накладные расходы, которые имеет JQuery
Клиентский/серверный язык	Обычно используется на стороне клиента	Обычно используется на стороне клиента	Обычно используется на стороне клиента
Объем файлов	Легкий по сравнению с другими библиотеками JavaScript	Легкий по сравнению с другими библиотеками JavaScript	Помимо необработанного JavaScript, он тяжелее, чем JQuery
Способ подключения	1. Может вставлен в теги <script> и </script> внутри HTML. 2. Загрузки с сайта jQuery.com. 3. Включена из CDN	1. Для одностраничного приложения. Следует использовать Create React App и npm: npm install -g create-react-app cd hello-world npm start.	1.Код необходимо писать внутри тега <script> и </script> в HTML. Не нужно импортировать/добавлять библиотеки. 2. Подключение отдельного файла со скриптом следующим образом: <script src="/my/script.js">

Критерий	jQuery	React.js	JavaScript
		2. Добавление React к уже существующему приложению: npm install --save react. 3. Использование CDN	</script>
Синтаксис	$ – знак для определения jQuery. Селектор – запрос для поиска элементов HTML. Action – действие JQuery, которое нужно выполнить	Применяется синтаксис JSX, который воспринимается как обычный HTML	Не существует специальных символов JavaScript. Мы можем просто начать писать код JavaScript в теге script в HTML

Исходя из таблицы, для практического сравнения выбраны следующие критерии:

– простота использования;
– совместимость;
– длина кода;
– скорость написания и обработки кода;
– объем файлов.

Литература

1. Карышев А.А., Афанасьев В.Р. Разработка web-приложения для автоматизированной генерации документов на основе docx-шаблонов // Изв. ТулГУ. Технические науки. – 2017. – Вып. 5. – С. 294–296.
2. Сравнение производительности JS-библиотек [Электронный ресурс]. – Режим доступа: https://habr.com/company/mailru/blog/188254/ (дата обращения: 21.12.2018).
3. 45% of the popular websites use a javascript framework [Электронный ресурс]. – Режим доступа: http://elie.im/blog/web/45-of-the-popularwebsites-use-a-javascript-framework/ (дата обращения: 22.12.2018).
4. Duckett J., Ruppert G., Moore J. JavaScript & jQuery: interactive front-end web development // Indianapolis, IN: John Wiley & Sons, 2014. – 622 p.
5. Geest J., Ettema M. Comparison of JavaScript libraries // Bibliotheek der R.U. – 2011. – P. 107–128.

Шигарева Ксения Николаевна
Год рождения: 1997
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № P41601
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: kseniy__5869@mail.ru

Готская Ирина Борисовна
Год рождения: 1957
Университет ИТМО, факультет программной инженерии
и компьютерной техники, д.педагог.н., профессор
e-mail: iringot@mail.ru

УДК 004

ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ДИЗАЙНА WEB-ИНТЕРФЕЙСОВ С ИСПОЛЬЗОВАНИЕМ ПОДХОДА КОГНИТИВНОЙ ПСИХОЛОГИИ

Шигарева К.Н.
Научный руководитель – д.педагог.н., профессор Готская И.Б.

В работе рассмотрена проблема влияния когнитивных процессов на проектирование дизайна web-интерфейсов, представлены результаты анализа зарубежных и отечественных исследований данной проблемы, а именно, основные характеристики, которыми должен обладать успешный и удобный интерфейс, а также выявлены особенности влияния когнитивных процессов на восприятие пользовательского интерфейса.
Ключевые слова: проектирование интерфейса, когнитивные процессы, дизайн, восприятие, пользовательский web-интерфейс.

Взаимодействие пользователя с интерфейсом, его поведение на сайте диктует новые требования к проектированию web-интерфейсов. В процессе проектирования и разработки интерфейсов важно хорошо разбираться в психофизиологии человеческого восприятия, так как когнитивные процессы пользователя оказывают непосредственное влияние на процесс его взаимодействия с интерфейсом.

Чрезвычайно сложно сделать web-интерфейсы доступными для самого широкого круга пользователей, особенно учитывая их ограничения в восприятии, управлении и понимании. Поэтому, чтобы создать интерфейс, соответствующий принципам юзабилити, необходимо знать психологию восприятия человеком компьютера, так считает Т.В. Кишкурно [1]. Разработчики, работая в тандеме с дизайнерами, создают новые оптимизированные методы, которые направлены на то, чтобы обеспечить эффективное и удобное взаимодействие пользователя с web-интерфейсом. Но однозначно подтвержденных научными исследованиями правил по созданию юзабилити интерфейсов с учетом когнитивных особенностей пользователей в настоящее время нет, что определяет актуальность темы научной работы.

Сегодня уже существуют некоторые информационные технологии, которые, так или иначе, способствуют решению проблемы по созданию юзабилити интерфейсов для различных категорий пользователей – это каскадные таблицы стилей, облака тегов и т.д. Однако такие технологии решают только техническую проблему создания пользовательских интерфейсов. В то время как глобальная цель пользовательского интерфейса – предоставить пользователю

возможность удобно управлять компьютером, с которым он взаимодействует, и получить обратную связь для эффективного выполнения задач. В связи с этим успешный пользовательский интерфейс должен быть интуитивно понятным, эффективным и удобным.

Дизайн пользовательского интерфейса, являющийся неотъемлемой частью продукта, представляет собой сложный проект с когнитивной психологией, определенными акцентами, лингвистикой и т.д. Существуют три принципа дизайна пользовательского интерфейса: настройка интерфейса под контролем пользователя, снижение нагрузки на память пользователя, поддержание согласованности интерфейса [2].

Так, Д. Гибсон [3], Р. Арнхейм [2] и другие выделяют основные качества и характеристики создания успешного интерфейса:

1. ясность. В интерфейсе не должно быть двусмысленности, что достигается с помощью языка, иерархии и метафор для визуальных элементов;
2. четкость. Избыточность компонентов интерфейса значительно осложняет поиск, поэтому задача разработчиков и дизайнеров – это создание лаконичного и одновременно понятного интерфейса;
3. знакомство. Использование реальных метафор для передачи смысла при первом знакомстве пользователя с интерфейсом;
4. отзывчивость. Обеспечение обратной связи с пользователем об успешности обрабатываемого ввода данных;
5. прощение. Отсутствие наказания пользователей за их ошибки, предоставление средств для их исправления;
6. минимализм. Сведение новизны к минимуму. Тем не менее, основные элементы пользовательского интерфейса могут быть трудны для анализа, особенно на начальных этапах разработки web-сайта. Без определенных ключевых элементов пользовательский интерфейс может стать слишком сложным, а эффективность сайта может снизиться.

Для создания интерфейсов необходимо учитывать, как пользователи будут воспринимать действия и поведение с интерфейсом, какие знания им понадобятся для интерпретации и реагирования [4]. Поскольку когнитивная наука получила огромное понимание механизмов и процессов, лежащих в основе восприятия, а именно: памяти, внимания, категоризации, принятия решений, можно предположить, что такие выводы очень актуальны для практических вопросов проектирования.

Нынешние точки зрения на восприятие в когнитивной психологии склонны фокусироваться на конкретных действиях, в которых человек замечает и понимает стимулы чувств, и как они влияют на производительность. Учитывая когнитивные навыки пользователей и особенности взаимодействия, дизайнеры сайтов и программисты пытаются сделать технические приложения более удобными для пользователя.

Иногда дизайн становится слишком сложным для понимания. Пользователь не может понять представленную информацию и ему нужно приложить дополнительные усилия, это означает, что когнитивная нагрузка увеличивается. Когнитивная нагрузка – это количество энергии, используемой мозгом при выполнении любой задачи, или количество умственных ресурсов, необходимых для работы системы [3]. Есть много причин, которые ответственны за увеличение когнитивной нагрузки, вот несколько из них:

– нерелевантные изображения или неправильная типографика способны увеличить когнитивную нагрузку;
– отсутствие ясности, когда пользователь достигает цели после множества ненужных кликов;
– длинная форма, которую необходимо заполнить, и большинство полей являются обязательными;
– наличие множества вариантов и выбор одного из них;
– отсутствие ожидаемого результата.

Снять когнитивную нагрузку очень сложно, но нужно постараться ее минимизировать. Выделяют некоторые законы, которые помогают понять психологию человека, минимизировать когнитивную нагрузку и улучшить пользовательский опыт [3].

1. Закон Фитта – «легче и быстрее ударить большую цель ближе к вам, чем меньшую цель дальше от вас». В дизайне размер объекта и расстояние от того места, где мы хотим взаимодействовать с этим объектом, имеют большое значение. Этот закон помогает дизайнерам понять, как распределять задачи по категориям и как отдавать им приоритет при соответствующем размещении.
2. Гештальт-законы – данные законы трактуют, как различные элементы воспринимаются человеческим зрением при их объединении определенным образом или в различном порядке. Они определяют, как привлечь внимание к важным элементам или как создать впечатление баланса и стабильности.
3. Закон сходства заключается в том, что отдельные объекты, которые кажутся похожими (с точки зрения формы, размера, цвета или любых других конкретных характеристик) друг на друга, пользователи воспринимают их, как связанные.
4. Закон продолжения объясняет то, как мозг воспринимает визуальные линии или продолжение элементов. Так, человеческий глаз следует за прямой или изогнутой линией и предпочитает видеть соединенную и непрерывную фигуру.
5. Закон закрытия говорит о том, что когда объект неполон или пространство не полностью закрыто, то мозг часто игнорирует противоречивую информацию и сам заполняет пробел.
6. Закон близости указывает, что элементы, которые находятся близко друг к другу, имеют тенденцию восприниматься как группу или единое целое.
7. Закон фигуры и земли объясняет отношения между объектом и его окружением. От того, как человеческий глаз различает объект (фигуру) от окружающей его области (земли), зависит от формы, размера, цвета и контрастности объекта.
8. Закон Хика – «увеличение количества вариантов увеличит время принятия решений». Закон Хика помогает проектировщикам распределять по категориям, облегчать навигацию и доступный пользовательский поток. Это экономит время и энергию пользователей, а также улучшает принятие решений.
9. Закон Миллера – «средний человек может хранить 7 (+/–2) номеров предметов в своей рабочей памяти». Пользователи запоминают информацию порциями. Разделение на части – это процесс группировки информации, связанной признаками восприятия. Поэтому для лучшего восприятия нужно разбивать контент и сохранять четкие визуальные иерархии с наиболее важной информацией в группе, чтобы пользователь мог легко принять информацию и запомнить важные вещи. Это также улучшает способность пользователя понимать.

Учитывая законы по снижению когнитивной нагрузки, выделяют некоторые общие рекомендации для создания хорошего пользовательского интерфейса [5]:

– общие элементы должны работать предсказуемо, чтобы пользователи могли неосознанно использовать их повсюду;
– поддержка высокой доступности;
– создавать простые интерфейсы, где у каждого элемента своя цель;
– уважение взглядов пользователя и внимание к макету, фокусирование на иерархии и удобочитаемости;
– выравнивание, т.е. минимизация количества выравнивающих линий;
– привлечение внимания к ключевым функциям, используя цвет, яркость, контраст;
– разумное количество цветов и оттенков;
– минимизация количества действий для выполнения задач, но фокус на одной главной функции на странице. Упрощение сложных задач, используя прогрессивное раскрытие;
– расположение элементов управления рядом с объектами, которые пользователи хотят контролировать;

– заблаговременный просмотр значений по умолчанию для того, чтобы уменьшить нагрузку на пользователя (например, формы предварительного заполнения).

Таким образом, взаимодействие между цифровым продуктом является неотъемлемой частью нашей жизни, и очень важно, чтобы продукты могли понимать поведение человека. Требуется большое количество усилий, чтобы создать интуитивно понятный дизайн, но в дальнейшем это сократит время обучения, снизит разочарование и повысит эффективность. Зрительное восприятие важно для когнитивных процессов, и между ними существует взаимосвязь. Когнитивные процессы обрабатывают информацию, формирующую зрительную систему, поэтому технические приложения стараются создавать более удобными, понятными и эффективными для пользователя, не пренебрегая эстетическими принципами.

Литература

1. Кишкурно Т.В., Брусенцова Т.П. Использование принципов юзабилити для оптимизации процесса восприятия экранного пространства [Электронный ресурс]. – Режим доступа: https://elib.belstu.by/bitstream/123456789/23641/1/Kishkurno_Ispol%27zovanie.pdf (дата обращения: 04.01.2019).
2. Arnheim R. Art and Visual Perception. – California: University of California Press, 1974. – 508 p.
3. Gibson J. The Ecological Approach to Visual Perception. – N.Y.: Taylor & francis Group, 1988. – 464 p.
4. Ушакова Т.Н., Холодная М.А., Чудова Н.В. Психология высших когнитивных процессов. – М.: ИП РАН, 2004. – 304 с.
5. Douglas I., Liu Z. Human – Computer Interaction Series: Global Usability. – London: Springer-Verlag, 2011. – P. 110–112.

Шишкин Роман Вячеславович
Год рождения: 1994
Университет ИТМО, факультет программной инженерии
и компьютерных технологий, студент группы № Р4272
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: rommmash@yandex.ru

Локалов Владимир Анатольевич
Год рождения: 1959
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.педагог.н., доцент
e-mail: localov@gmail.com

УДК 004.51

ФАКТОРЫ ВЛИЯНИЯ НА ВОСПРИЯТИЕ ЦВЕТОВОЙ СТРУКТУРЫ КОМПЬЮТЕРНОГО ИЗОБРАЖЕНИЯ

Шишкин Р.В., Локалов В.А.
Научный руководитель – к.педагог.н., доцент Локалов В.А.

Работа посвящена исследованию восприятия цветовой структуры компьютерного изображения. Введено понятие «субъективная цветовая структура», и предложена система факторов, влияющих на процесс цветового восприятия.
Ключевые слова: цветовая структура, восприятие цвета, проектирование интерфейсов.

Выбор цветового решения цифрового продукта – важнейший этап его проектирования. Цвет в интерфейсе может управлять вниманием пользователя, помогать распознавать элементы интерфейса, передавать смысл индикаторов в сложных профессиональных системах, а также использоваться для визуальной группировки сходных объектов. При разработке любого интерфейса его цветовое решение необходимо ориентировать на субъективную цветовую структуру. Назовем совокупность значимых цветовых единиц, которые человек замечает в первую очередь при восприятии изображения. Функциональные цвета в интерфейсе должны коррелировать с этой структурой, иначе в противном случае важные интерфейсные элементы будут упущены пользователем [1–4].

Специальные компьютерные средства, использующие методы кластеризации, могут однозначно определить, какие цвета присутствуют в изображении. Однако полученные кластеры не отображают субъективную цветовую структуру изображения. Это связано с особенностями восприятия и многочисленными факторами, влияющими на этот процесс.

Важнейшим фактором, обуславливающим характеристики восприятия компьютерного изображения, является задача, которую решает пользователь. С изменением цели и условий компьютерного взаимодействия, меняется механизм рассматривания объекта и рисунок окуломоторной активности. При этом меняется не только содержание значимых элементов (что и где воспринимается), но и требования к их восприятию (насколько точно и дифференцированно). Другими словами, если перед пользователем ставится задача, заключающаяся в поиске цветовых объектов, то процесс восприятия является направленным и регулируемым.

Не менее важным является фактор сюжетности. При восприятии сюжетного изображения человеку бывает тяжело абстрагироваться от сущности изображенных объектов, таким образом, воспринимается не цвет, а объект. Такое явление часто называют «интеллектуальным реализмом». Сюжетное изображение может содержать антропогенные и геометрические направляющие (жесты рук, взгляд), которые, в свою очередь, могут также оказывать влияние на восприятие цветовой структуры. Также влиять могут стереотипы рассматривания. Это и аналогичное процессу чтения рассматривание слева-направо и рассматривание по диагонали.

Третьим по значимости фактором является дифференциальные характеристики цветовых пятен (количество цвета, форма пятен яркость, светлота, тон). Сложно не заметить цвет, который занимает ½ площади всей поверхности изображения. Чем проще и локальнее пятно, тем быстрее оно бросится в глаза. Цветные элементы в форме круга, многоугольника или других простых форм всегда будут преобладать над аморфной формой пятна с рваными краями, так как их конструкция наиболее понятна смотрящему. В большинстве случаев форма преобладает над цветом, вследствие чего интерес к цвету отходит на второй план. На характеристиках цвета базируются основные принципы теории визуального восприятия: контраста, акцента, которые отчасти определяют подход к восприятию субъективной цветовой структуры.

Набор влияющих факторов зависит от типа изображения, можно нивелировать значимость одного из них. Например, используя в качестве стимульного материала абстрактные изображения, и не ставя перед пользователем конкретной задачи, исключим из зоны влияния фактор сюжетности. Но следует заметить, что есть группа факторов, влияние которых исключить из системы невозможно – визуальные иллюзии, все, что связано с индивидными и личностными характеристиками смотрящего.

Необходимо понимать, что цвет постоянно нас обманывает, но почти никогда нельзя увидеть отдельный цвет вне связи с другими цветами. В зависимости от детализации картинки, группировки цветов порождают иллюзии, которые заставляют нас видеть и считывать совершенно другие цвета, нежели те, которые перед нами на самом деле. Каждый человек имеет свои личностные детерминанты восприятия, опираясь на свои особенности: физиологические, психологические, культурные-социальные и т.д. Мы соотносим цвет с определенными предметами, явлениями, эмоциями, причем некоторые из этих связей становятся очень прочными в силу своей повторяемости и очевидности: желтый – солнце, красный – возбуждение.

Существует гипотеза, что люди видят и воспринимают в изображении преимущественно свои любимые цвета. Каждый человек имеет свои вкусы, склонность к одним цветам и антипатию к другим. Цветовые предпочтения могут быть обоснованы цветотипами человека, введенным М. Люшером. Причем, это не означает, что человеку с красным типом поведения нравится только красный цвет. К тому же, со временем отношение человека к тому или иному цвету может меняться. Для определения роли влияния цветовых предпочтений был проведен эксперимент, который проверил эту гипотезу.

Основной задачей эксперимента было сравнение заметных и предпочтительных цветов, определение зависимости заметности цветов в изображении от их привлекательности.

При проведении эксперимента перед испытуемыми ставились три задачи:

1. распределить четыре картинки в порядке эстетической привлекательности гамм (от более приятной до менее приятной);
2. из каждой картинки выделить цвета в порядке заметности;
3. пройти тест М. Люшера и назвать любимые цвета.

В эксперименте приняли участие 16 испытуемых в возрасте от 22 до 27 лет, не имеющих нарушений цветового зрения.

Проанализировав результаты, можно сделать вывод, что задачу чтения цветовой структуры респонденты решают по-разному (рис. 1). Алгоритм выделения цветовой

структуры разнообразен и неоднозначен, так как понятие «заметности» цвета для каждого человека имеет свой смысл.

Рис. 1. Вид цветовой структуры (а); примеры субъективных цветовых структур у разных респондентов (б–г), полученных при восприятии изображения

Цветовые предпочтения определялись двумя способами: тестом Люшера и словесным опросом после проведения эксперимента. Только у 2 из 16 испытуемых любимые цвета совпали с предпочтительными. Это может говорить о незначительной взаимосвязи предпочтительных цветов и заметных, и низкой степени существенности цветовых предпочтений как фактора влияния на восприятие цветовой структуры. Наличие любимых цветов в изображениях также не повлиял на привлекательность изображения в целом (приоритет в расстановке).

Обобщая все вышесказанное, получена система факторов, влияющих на процесс цветового восприятия (рис. 2), где отражена степень значимости того или иного фактора.

Рис. 2. Система факторов влияния на восприятие цветовой структуры компьютерного изображения

Цвет – самое относительное визуальное средство, но функционально он является самым мощным помощником. Сосредоточив внимание на исследовании его восприятия, можно более грамотно и эффективно использовать его в решении задач.

Литература

1. Барабанщиков В.А., Жегалло А.В. Айтрекинг: методы регистрации движений глаз в психологических исследованиях и практике. – М.: Когито-Центр, 2014. – С. 117–119.
2. Аббасов И.Б. Визуальное восприятие: учебное пособие. – М.: ДМК Пресс, 2016. – 136 с.
3. Альберс Дж. Взаимодействие цвета / Пер. с англ. Д. Халиковой. – М.: Колибри, Азбука-Аттикус, 2017. – 216 с.
4. Гавричкова Е.С., Петрова О.А. Влияние цвета на восприятие человека // Актуальные проблемы авиации и космонавтики. – 2012. – Т. 2. – № 8. – С. 300–301.

Ярош Станислав Вадимович
Год рождения: 1993
Университет ИТМО, факультет программной инженерии
и компьютерной техники, студент группы № Р4260
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: stanislav.yarosh@gmail.com

Шалобаев Евгений Васильевич
Год рождения: 1947
Университет ИТМО, факультет программной инженерии
и компьютерной техники, к.т.н., доцент
e-mail: shalobaev47@mail.ru

УДК 004.51

АНАЛИЗ ПРОГРАММНЫХ СРЕДСТВ СОЗДАНИЯ ИНТЕРФЕЙСНОЙ АНИМАЦИИ

Ярош С.В.
Научный руководитель – к.т.н., доцент Шалобаев Е.В.

В работе проведен анализ проблем, решаемых посредством интерфейсной анимации, дополнен анализ программных средств создания интерфейсной анимации. Выявлены причины, по которым следует менять способ создания интерфейсной анимации. Определены преимущества и ограничения самых популярных программных средств интерфейсной анимации.
Ключевые слова: интерфейсная анимация, пользовательский интерфейс, пользовательское взаимодействие, прототипирование, программные средства, дизайн.

В настоящее время роль дизайна в разработке цифрового продукта (digital product) крайне высока, – это в первую очередь связано с огромной конкуренцией. В связи с этим дизайнеры вынуждены совершенствовать свои навыки, изучать новые методы и технологии (инструменты) для эффективного достижения поставленных задач.

Технологии стремительно развиваются, появляются новые подходы, методы и инструменты для создания интерфейсной анимации. Появляется много новых программных решений [1], которые предлагают инструменты, нацеленные на удовлетворение потребностей современного дизайнера. Сегодня зачастую над цифровым продуктом работают целые команды. Благодаря новым программным средствам процесс передачи готового дизайна разработчикам, а также коммуникация предлагаемого интерфейсного решения членам команды или руководству становится легче.

В современном мире то, как выглядит цифровой продукт, и насколько хорошо спроектировано пользовательское взаимодействие с интерфейсом, имеет огромное значение. Среди компаний, которые разрабатывают программные средства для дизайнеров интерфейсов, существует тенденция по созданию инструмента, который позволяет дизайнеру реализовывать полный спектр работ, связанных с проектированием интерфейса сайта, системы или мобильного приложения. В связи с этим можно сделать вывод, что в целом программное обеспечение развивается таким образом, чтобы оптимизировать работу дизайнера, ускорить процесс проектирования, а также уменьшить разрыв между проектированием интерфейса и его разработкой.

При проектировании интерфейсов дизайнеры сталкиваются с задачами разного рода,

следует внимательно подходить к выбору инструмента, чтобы решить поставленную задачу наиболее эффективно.

Качественная анимация выполняет определенную функцию в интерфейсе, воспринимается пользователем как неотъемлемая часть взаимодействия с интерфейсом, а также привносит характер, присущий данному цифровому продукту [2].

Дизайн – это процесс, состоящий из этапов. Прототипирование (создание интерфейсной анимации) – это не часть дизайн-процесса, а сам процесс дизайна [3]. При проектировании интерфейсов, дизайнеры создают интерактивные прототипы на каждой итерации цикла разработки цифрового продукта по двум причинам:

– получить обратную связь по предлагаемому решению проблемы (дизайну) от пользователей, которые участвуют в тестировании, а также от членов команды;
– передать идею более эффективно, чтобы получить одобрение на разработку от руководства.

Выделяют два вида прототипирования:

1. создание прототипа интерфейса для понимания:
   – исследование идей;
   – поиск эффективного решения;
   – проверка гипотез;
   – получение обратной связи от членов команды.

Дизайнер создает большое количество простых прототипов, которые решают одну и ту же проблему, настраивает анимацию (выбирает функцию сглаживания), ищет лучший способ коммуникации обратной связи пользователю. После того как дизайнер находит решение проблемы, он переходит к созданию более детального прототипа;

2. создание прототипа для коммуникации (для членов команды и руководства):
   – «продажа» идеи руководству;
   – создание сложного прототипа (детально проработанного) для дальнейшего использования в проектах в качестве отправной точки при решении новых задач;
   – объяснение предлагаемого решения коллегам;
   – использование его в качестве маркетинговой составляющей для привлечения клиентов.

Также выделяют три причины, по которым следует переходить от создания простого прототипа к сложному (детально проработанному):

1. при демонстрации своего решения коллегам или руководству в виде спецификаций и диаграмм, дизайнера никто не понимает, поэтому следует упростить подачу решения – сделать анимированный прототип;
2. члены команды игнорируют идею, не относятся к ней серьезно, поэтому нужно обратить на нее внимание и вызвать обсуждение для получения необходимой обратной связи;
3. нужно донести видение будущего функционала продукта до членов команды, распространить идею, заинтересовать и начать диалог.

Скорость является важным фактором в разработке цифровых продуктов. Прототипирование – это способ поиска эффективного решения проблемы. Чем быстрее происходит данный поиск, тем лучше для продукта: дизайнеры быстрее и чаще получают обратную связь, итерации в разработке становятся короче, освобождается время для решения других проблем, продукт развивается быстрее, обгоняя конкурентов [4]. Для быстрого создания идей и получения обратной связи создаются простые прототипы, не требующие много времени на их создание.

Существуют инструменты, которые подходят для создания визуальной составляющей, но они не обладают возможностями создания интерактивности (бумажный скетч, Keynote, After Effects). Keynote и After Effects позволяют экспортировать созданную анимацию в формате видео или gif. Данные инструменты можно использовать только для демонстрации, они не подходят для тестирования предлагаемого решения – это являются большим недостатком при разработке цифровых продуктов.

Sketch, XD, Figma предоставляют все необходимые инструменты для создания визуальной составляющей интерфейса, дают возможность создавать интерактивные прототипы, соединяя экраны интерфейса между собой в заданной последовательности посредством «горячих» областей (hotspot). Также можно применять стандартную анимацию переходов от одного экрана к другому. Недостатком данных программных средств является отсутствие полноценного контроля над анимацией элементов управления интерфейсом.

Marvel и InVision являются онлайн-сервисами, главной задачей которых, является быстрое создание простого интерактивного прототипа для веб или мобильных устройств посредством «горячих» областей. Главными недостатками данных программных средств являются: простой инструментарий создания визуальной составляющей интерфейса, дизайнеру приходится создавать дизайн интерфейса в других программах; прототипирование происходит в браузере – нет возможности работать без Интернета.

Flinto и Principle позволяют создавать анимацию микровзаимодействия высокой детализации. В данных программах можно анимировать абсолютно любой элемент интерфейса, а также детально настраивать анимацию. Данные инструменты позволяют легко и быстро создать интерактивный прототип. При создании интерфейсной анимации используется Sketch для импорта готового дизайна интерфейса. Нет полноценного и удобного инструментария создания визуальной части интерфейса. Также данные инструменты требуют инвестиции времени для изучения их функционала.

Kite Compositor, Framer X и Origami позволяют создавать более сложную интерфейсную анимацию и интерактивные прототипы. В данных программных средствах есть инструменты для работы с камерой, жестами и другими сенсорами телефона, что может быть очень востребовано при решении определенных задач. Их недостаток – это сложность в освоении. Главные преимущества Kite Compositor, Framer X и Origami перед другими программными средствами – это возможность создания сложной логики взаимодействия, работа с реальными данными. Framer X идеально подходит для крупных проектов, в которых уже задействовано большое количество дизайнеров и разработчиков, а также для компаний, отделы дизайна и разработки которых постепенно увеличиваются, и возникает необходимость делиться элементами управления интерфейсом – компонентами [5].

InVision Studio предоставляет инструментарий для создания детальной анимации и прост в использовании. Главной особенностью является работа с анимацией элементов интерфейса посредством временной шкалы (timeline), что очень удобно. Также в данной программе есть все необходимые инструменты для создания визуальной части интерфейса.

При работе над цифровым продуктом дизайнеры используют не один инструмент, а несколько, что позволяет решать поставленные задачи более эффективно. Популярным набором инструментов является: Sketch, Principle, InVision.

Используя программные средства создания интерфейсной анимации, возникает потребность в управлении взаимодействием с интерфейсом посредством жестов и прокрутки (какие-то элементы на странице остаются на месте, другие можно прокручивать/пролистывать). Программные средства создания интерфейсной анимации должны иметь инструменты, позволяющие управлять:

- позицией элементов (направление движения элементов интерфейса);
- размером (показывает, на каком расстоянии находится объект, подчеркивает его важность, привлекает внимание пользователя);
- поворотом (показывает изменение в системе, например, обновление);
- размытостью (добавляет эффект глубины и иерархии, например, пользователь может видеть, что за размытым фоном что-то находится);
- прозрачностью (управление объемом, создание более плавных и естественных переходов);
- масштабом (фокусирует внимание пользователя на главном);
- цветом (привлекает внимание пользователя);
- перекрытием (дает понять, где находится 2D-объект в 3D-пространстве);

– функцией сглаживания (ускоряет или замедляет анимацию элемента).

Выбор правильного программного средства создания интерфейсной анимации существенно влияет на эффективность разработки и скорость внедрения новых решений. В таблице проведено сравнение самых популярных инструментов по критериям, выявленным из анализа задач, с которым сталкиваются дизайнеры цифровых продуктов. Для обозначения цены используются следующие обозначения: б – бесплатно; п – по подписке; л – единовременная лицензия.

Таблица. Сравнение программных средств создания интерфейсной анимации

Критерии	Keynote	Sketch	Figma	Adobe XD	InVision Studio	Flinto	Principle	After Effects	Kite Compositor	Origami Studio	Framer X	Marvel	InVision
Визуальная часть	+	+	+	+	+	+	+	+	+	+	+	+	+
Переходы	+	+	+	+	+	+	+		+	+	+	+	+
Детальная анимация					+	+	+	+	+	+	+		
Сложная логика									+	+	+		
Реальные данные		+	+	+					+	+	+		
Интернационализация		+							+		+		
Интерактивные компоненты									+	+	+		
Демонстрация на устройстве		+	+	+	+	+	+		+	+	+	+	+
Экспорт кода анимации								+	+	+	+		
Поддержка сенсоров						+			+	+	+		
Совместная работа			+									+	+
Работа в браузере			+									+	+
OS X	+	+		+	+	+	+	+	+	+	+		
Windows				+				+					
Цена	б	л	п	б	б	л	л	п	л	б	п	п	п

В работе проанализированы проблемы, решаемые посредством интерфейсной анимации. Выявлены причины, по которым следует менять подход к созданию интерфейсной анимации. Определены преимущества и ограничения программных средств создания интерфейсной анимации. В дальнейшем исследовании планируется составить практические рекомендации по выбору программного средства создания интерфейсной анимации для решения конкретных задач.

Литература

1. Palmor T. All the UX prototyping tools Ever [Электронный ресурс]. – Режим доступа: https://uxtools.co/blog/all-the-ux-prototyping-tools-ever/ (дата обращения: 04.01.2019).
2. Stanley P. Practical tips for great UI animation [Видеозапись]. – Режим доступа: https://youtu.be/LmXVxkWjLT8 (дата обращения: 04.01.2019).
3. Hu S. The prototyping process [Видеозапись]. – Режим доступа: https://youtu.be/l6AyYhNch-w (дата обращения: 04.01.2019).
4. Head V. Designing Interface Animation. – Rosenfeld Media, 2016. – 240 p.
5. Pelkov L. The New Framer X: Initial Impressions [Электронный ресурс]. – Режим доступа: https://www.smashingmagazine.com/2018/10/new-framer-x-initial-impressions/ (дата обращения: 04.01.2019).

СОДЕРЖАНИЕ

Направление «ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ»..... 4
Антонова А.Д. Художественная визуализация в среде VR 5
Бабарицкий П.А., Государев И.Б. Анализ и имплементация методов извлечения и обработки данных в вебе 8
Бабахин А.В., Государев И.Б. Сравнительный анализ CSS-методологий..... 13
Баев А.А., Государев И.Б. Моделирование системы рейтинга в многопользовательских играх на веб-платформе 16
Балаев А.К., Государев И.Б. Методологии рефакторинга унаследованного кода. Подготовительная фаза рефакторинга..... 18
Рущенко Н.Г., Барышева К.В. Анализ применения переиспользуемых элементов в современных системах диспетчерского контроля и сбора данных 20
Безбах Ю.И. Организация доступной среды для слабовидящих средствами дополненной реальности 23
Васильченко И.Д. Исследование особенностей UI/UX-дизайна mHealth-приложений.....26
Волков В.О. Модернизация локальной вычислительной сети мобильного узла связи на базе автомобиля камаз..... 30
Воробьев П.В., Махлай Д.О., Сопроненко Л.П., Чернева В.И. Использование виртуальной реконструкции при создании мультимедийной системы «Русский стиль»..... 35
Габдуллазянова З.Р. Проектирование интерфейсов для обучения детей с дислексией..... 39
Гарифуллина Г.Р. Оптимизация работы веб-приложений с помощью управления кэшированием 42
Геворкян А.А., Меженин А.В. Методы повышения эффективности поиска локаций.....45
Герасимов А. Анализ и проектирование модели данных современных дизайн-систем..... 49
Голубева В.А. Исследование дизайна мобильных приложений с учетом психологического восприятия цвета и формы..... 55
Демашов Д.С. Сравнительный анализ фреймворков на платформе Node.js..... 59
Дембелова А.М., Меженин А.В. Методы повышения качества 2D-компьютерной анимации 63
Деркунская С.А., Шуклин Д.А. Исследование тематических ситуаций динамической визуальной айдентики..... 68
Довгань А.Н. Реконструкция изображений, смазанных равномерно и неравномерно.....72
Дорофеева М.И. Влияние когнитивных процессов на дизайн веб-сайтов..... 78
Жмылёв С.А., Мартынчук И.Г., Киреев В.Ю. Архитектурное проектирование сервиса автоматического управления облачной инфраструктурой 82
Зиннатулин Ф.Ф., Шалобаев Е.В. Корпоративный портал как информационная модель предприятия 87
Зиновьев С.С., Меженин А.В. Методы оценки качества визуализации 90
Иванова П.В. Анализ интерактивных комиксов для детей..... 94
Калёнова О.В., Перл И.А. Сравнение подходов к обработке гетерогенных данных 98
Капутерка Э.А. Влияние анимированных переходов в графических интерфейсах на человеко-компьютерное взаимодействие..... 103
Кинцель Н.Б., Шаров Д.Д Иллюзии восприятия перспективы..... 107
Ключев В.А., Жовницкий В.А., Ястребов Е.Е. Низкоуровневое программирование аппаратного обеспечения и его место в современном учебном процессе 110

Жовницкий В.А., Ключев В.А., Ястребов Е.Е. Проблемы удаленного обновления программного обеспечения киберфизических систем 113
Ковалев В.В. Сравнительный анализ моделей обслуживания и развертывания облачных решений в вебе 118
Кондулукова Д.А. Повышение точности восстановления искаженного изображения путем определения параметров искажений по Фурье-спектру 123
Кореньков Ю.Д., Логинов И.П., Кузенкова Е.В. Применение декларативной модели описания целевых архитектур к существующим ISA 130
Кудрявых А.Ю. Сравнительный анализ использования монолитной и микросервисной архитектуры приложений 135
Кудряшов А.П., Меженин А.В. Методы оптимизации сцен в игровых компьютерных приложениях 139
Кузенкова Е.В., Кореньков Ю.Д., Логинов И.П. Сравнение сборщиков мусора в различных платформах 144
Кузнецова Н.В. Сравнительный анализ методологий разработки на платформе PHP: асинхронность в PHP 150
Кузьмин К.С. Метод обеспечения безопасности передачи данных при обновлении программного обеспечения мобильных устройств 154
Лавров А.В. Оценка погрешности восстановления частично перекрывающихся спектральных линий 158
Лаврова А.В., Спиридонова А.М., Лавров А.В. Интерфейс перемещения между музейными пространствами в системах виртуальной реальности 164
Лавская Л.В., Зенцова Е.С. Анализ тенденций применения UI/UX-технологий к созданию интерфейса мобильного приложения 167
Литвинов К.В. Проблема взаимодействия устройств в сфере интернета вещей 170
Максименко А.Е., Смолин А.А., Корпан Л.М. Вариативные шрифты как новая технология экранной типографики 172
Мариничев Н.А., Меженин А.В. Оценка качества захвата движения при использовании связки из двух и трех Microsoft Kinect 2.0 176
Машина Е.А. (Университет ИТМО), **Сушенцова М.В.** (Московский государственный университет им. М.В. Ломоносова), **Нелепко Л.Н.** (Санкт-Петербургский государственный университет), **Баркалов М.М.** (Университет ИТМО), **Бейлин М.Т.** (Университет ИТМО), **Орлова Л.Д.** (Университет ИТМО), **Григорьева Е.И.** (Университет ИТМО). Систематизация начального информационно-технологического контента, необходимого для создания системы удаленной информационной поддержки муниципальных дружин добровольной самопомощи PersonalMedHelper 179
Машина Е.А. (Университет ИТМО), **Сушенцова М.В.** (Московский государственный университет им. М.В. Ломоносова). Применение системы удаленной информационной поддержки добровольных муниципальных дружин доврачебной самопомощи PersonalMedHelper для повышения качества жизни жителей населенных пунктов в рамках программы устойчивого развития территорий 186
Мерзлякова Ю.В. Исследование дизайна инфографики для поддержки уроков в основной школе 191
Меркулова О.В. Сравнительный анализ JavaScript-фреймворков 194
Миннемуллин Б.М. Игрофицированное веб-приложение для дистанционного обучения школьников 198
Миронова М.М., Меженин А.В. Виды презентационного видео в театре 202
Молодкина А.А., Локалов В.А. Индивидуальные особенности восприятия визуальных напряжений компьютерного кадра 205

Молодцова В.С., Шуклин Д.А. Анализ программного обеспечения применения технологии дополненной реальности в информационном моделировании зданий (BIM)..... 209
Мосолова И.С. Методы вовлечения пользователей в виртуальную реальность..... 213
Никитинская В.М., Государев И.Б. Классификация инструментов генерации статических веб-сайтов..... 216
Носков И.И Обоснование выбора среды имитационного моделирования для исследования отказоустойчивых компьютерных сетей..... 220
Павлов Е.И. Проектирование образовательной веб-платформы «Цифровая грамотность»..... 226
Паршин Д.С., Учанева А.А. Особенности верстки веб-сайтов под мобильные устройства..... 230
Перес О.М.А. Визуализация и сегментация тропических циклонов в спутниковых изображениях с использованием техники Дворжака..... 234
Постникова П.И. Исследование положительных и отрицательных сторон мобильных приложений как инструментов коммуникации (на примере приложений «Госуслуги», «Сбербанк онлайн» и «Наш Санкт-Петербург)»..... 238
Прижимова Г.О. Исследование методов создания функциональной web-анимации на платформе CSS..... 241
Ростов Д.С., Готская И.Б., Государев И.Б. Исследование методов имплементации рендеринга клиентского интерфейса на стороне сервера..... 244
Румянцева А.Г. Анализ применения технологий дополненной реальности в музейном пространстве..... 247
Рунаева В.В., Государев И.Б. Особенности применения технологии виртуальной реальности в веб-браузере..... 249
Сенин А.Д., Государев И.Б. Применение реактивной парадигмы программирования в управлении состоянием веб-приложений..... 253
Сергиенко А.А., Рущенко Н.Г. Новый быстрый алгоритм восстановления смазанных изображений..... 257
Смогоржевский Г.А., Меженин А.В. Методы генерации городской инфраструктуры в медийных приложениях..... 263
Степанов М.Ю. Модель прогнозирования начального уровня испытуемого в электронных системах тестирования..... 266
Ткешелашвили Н.М., Махновец В.А., Ткешелашвили А.М., Клименков С.В. Модуль конвертации документов для системы анализа текстов на естественном языке..... 271
Трушина Ю.А., Меженин А.В. Методы уменьшения искажений при запекании карт нормалей..... 274
Урываев К.А. Обзор методов повышения качества прогнозирования с использованием искусственных нейронных сетей..... 278
Фишман В.О., Меженин А.В. Дизайн интерфейсов автомобильных мультимедиа систем..... 282
Фролов А.И. Анализ алгоритма градиентного бустинга для целей прогнозирования..... 285
Хорошавина А.С. Подход к разработке метода интеллектуального анализа данных для автоматизированной проверки знаний по английскому языку..... 289
Челокьян Л.А. Создание интерфейсов для людей с ограниченными возможностями..... 293
Шайхетдинова А.Р. Исследование методов поискового маркетинга в мобильной выдаче . 297
Шаповалова Д.В. Анализ научных исследований по проблеме создания 2D- и 3D-анимации в мультипликации..... 300

Швалева Д.С., Государев И.Б. Критерии сравнения методологий управления объектной моделью веб-документа..... 305
Шигарева К.Н. Особенности проектирования дизайна web-интерфейсов с использованием подхода когнитивной психологии 309
Шишкин Р.В., Локалов В.А. Факторы влияния на восприятие цветовой структуры компьютерного изображения..... 313
Ярош С.В. Анализ программных средств создания интерфейсной анимации..... 316

АЛЬМАНАХ НАУЧНЫХ РАБОТ МОЛОДЫХ УЧЕНЫХ УНИВЕРСИТЕТА ИТМО
Том 1

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Дизайн обложки Н.А. Потехина
Зав. РИО Н.Ф. Гусарова
Редактор Л.Н. Точилина
Подписано к печати 10.10.2019
Заказ № 4236
Тираж 100 экз.